Image Synthesis

EECS 442 – Jeong Joon Park

Winter 2024, University of Michigan

Administrivia

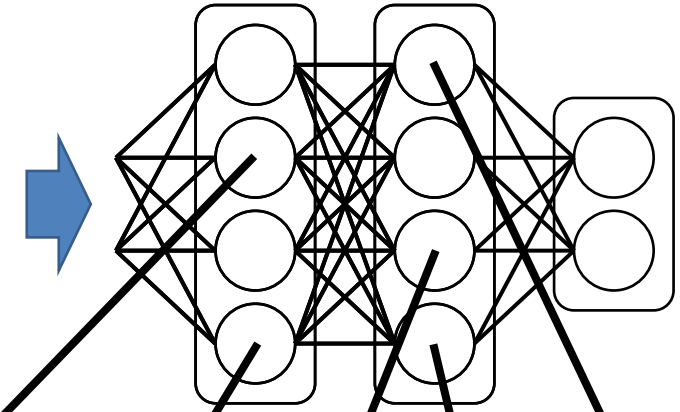
- Mid-term practice exam up – check Piazza
- The real one will cover similar topics
- And easier than the practice one
- Bring a pen – no cheat sheet, laptop, ipad, etc
- Will be seated randomly

Administrivia

- Project proposal due 27th
- We won't give feedback to the proposals
- Instead, come to OH or Discussions to receive feedback

Recap on Self-Supervised Learning

ImageNet + Deep Learning



Beagle

Materials?

Parts?

Pose?

Geometry?

Boundaries?

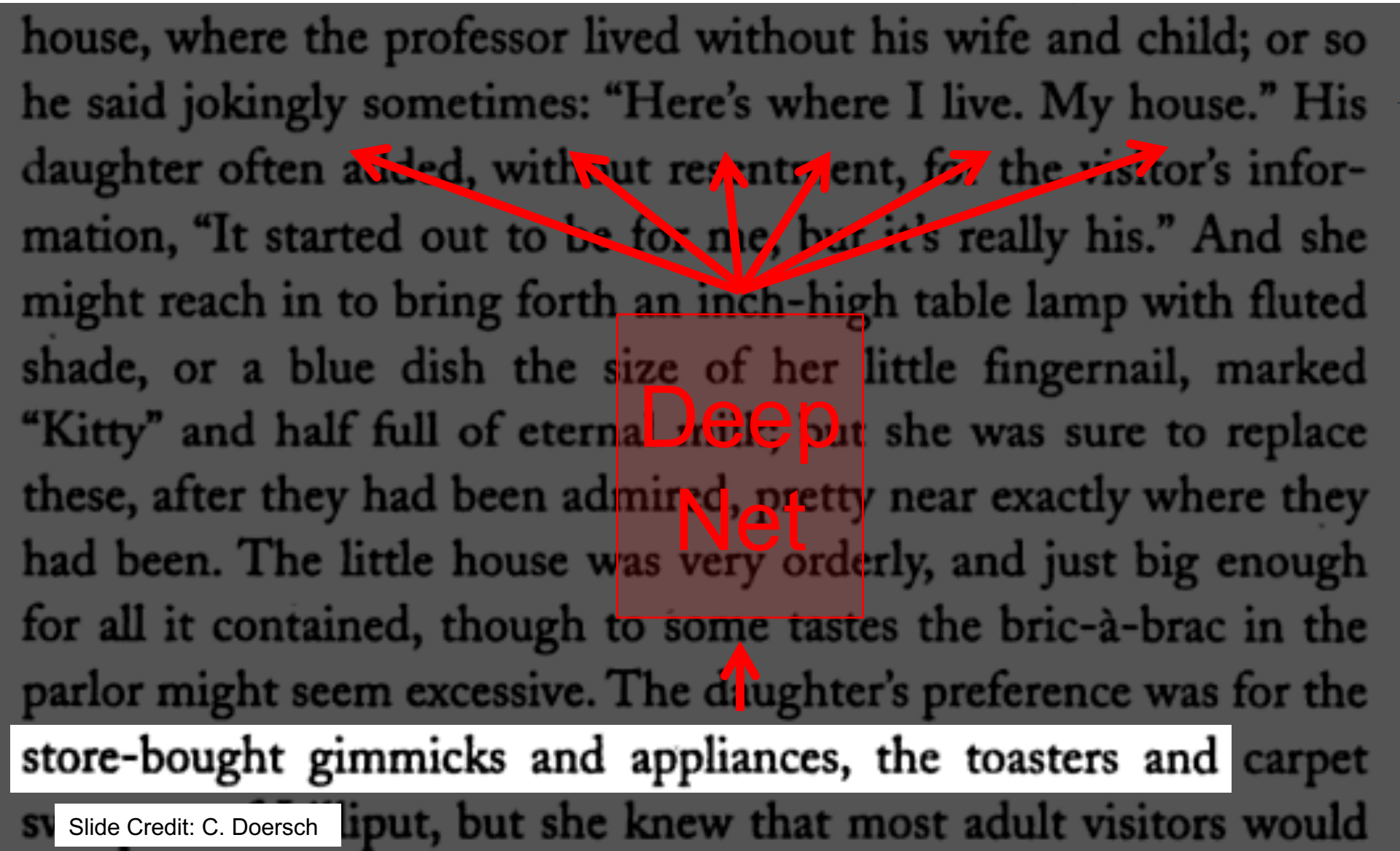
Do we even need semantic labels?

Do we need this task?

Slide Credit: C. Doersch

Context as Supervision

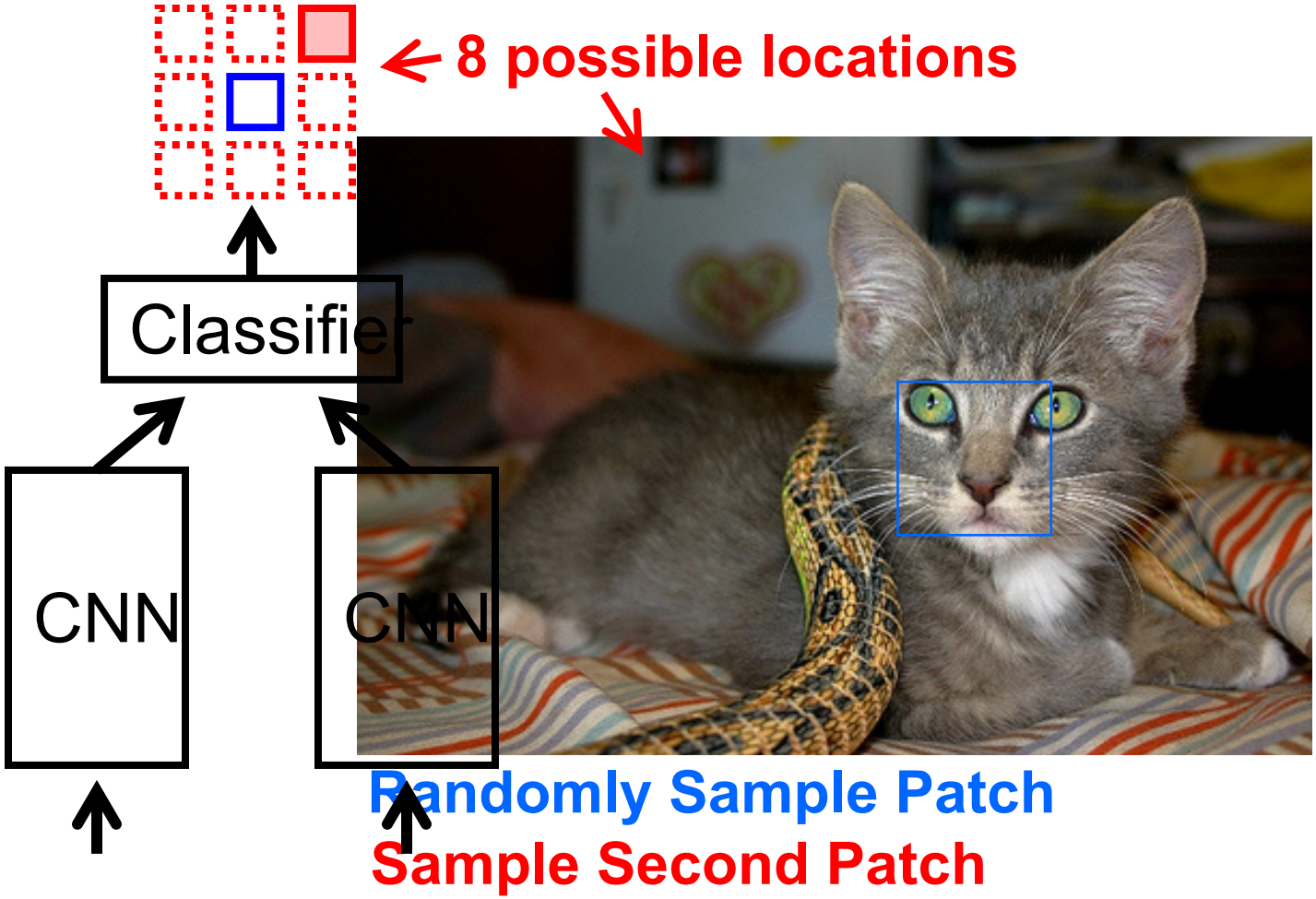
[Collobert & Weston 2008; Mikolov et al. 2013]



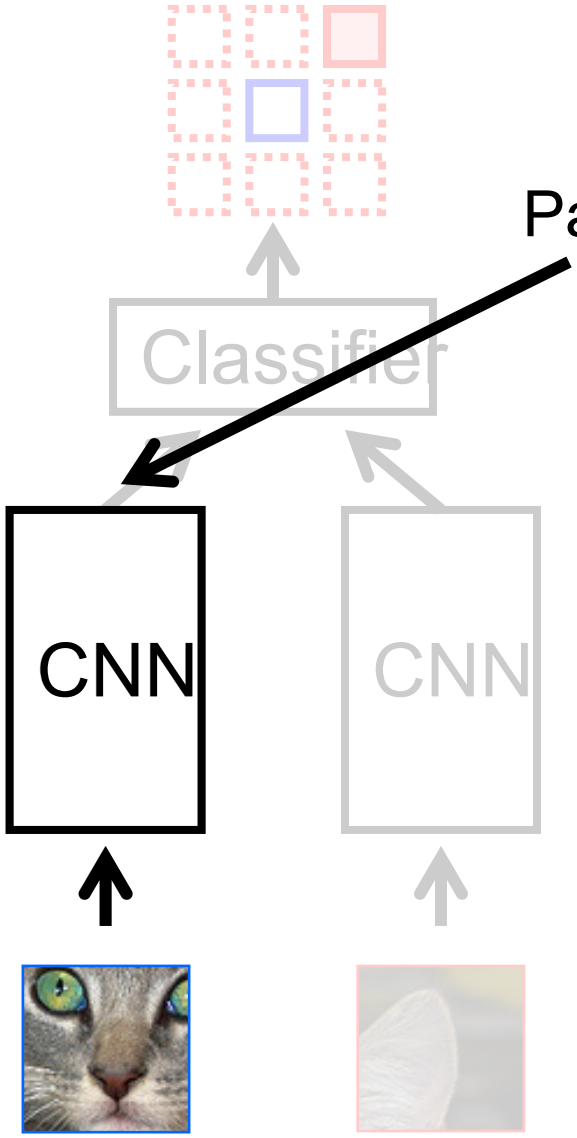
house, where the professor lived without his wife and child; or so he said jokingly sometimes: "Here's where I live. My house." His daughter often added, without resentment, for the visitor's information, "It started out to be for me, but it's really his." And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked "Kitty" and half full of eternal dimes, but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter's preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers Lilliput, but she knew that most adult visitors would

Slide Credit: C. Doersch

Relative Position Task



Slide Credit: C. Doersch



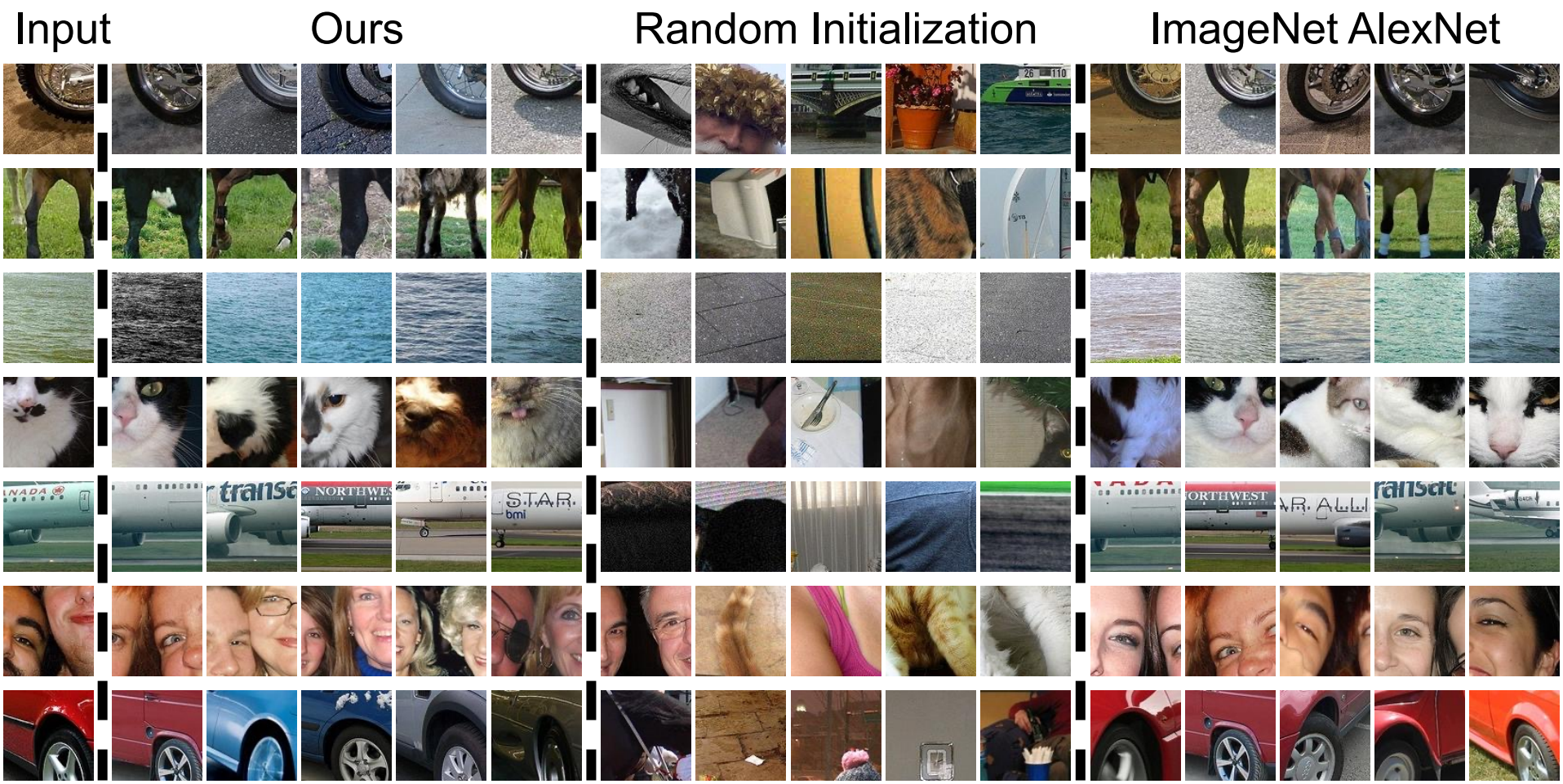
Input

Nearest Neighbors

Note: connects ***across*** instances

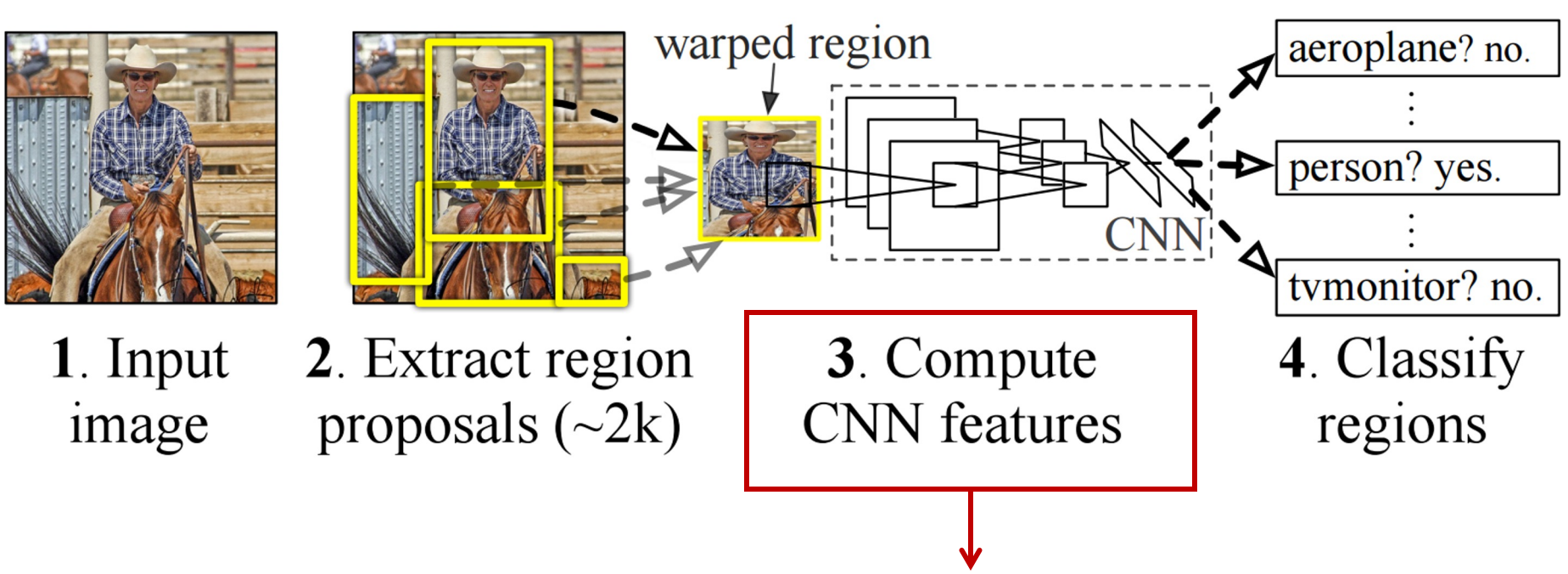
Slide Credit: C. Doersch

What is learned?



Slide Credit: C. Doersch

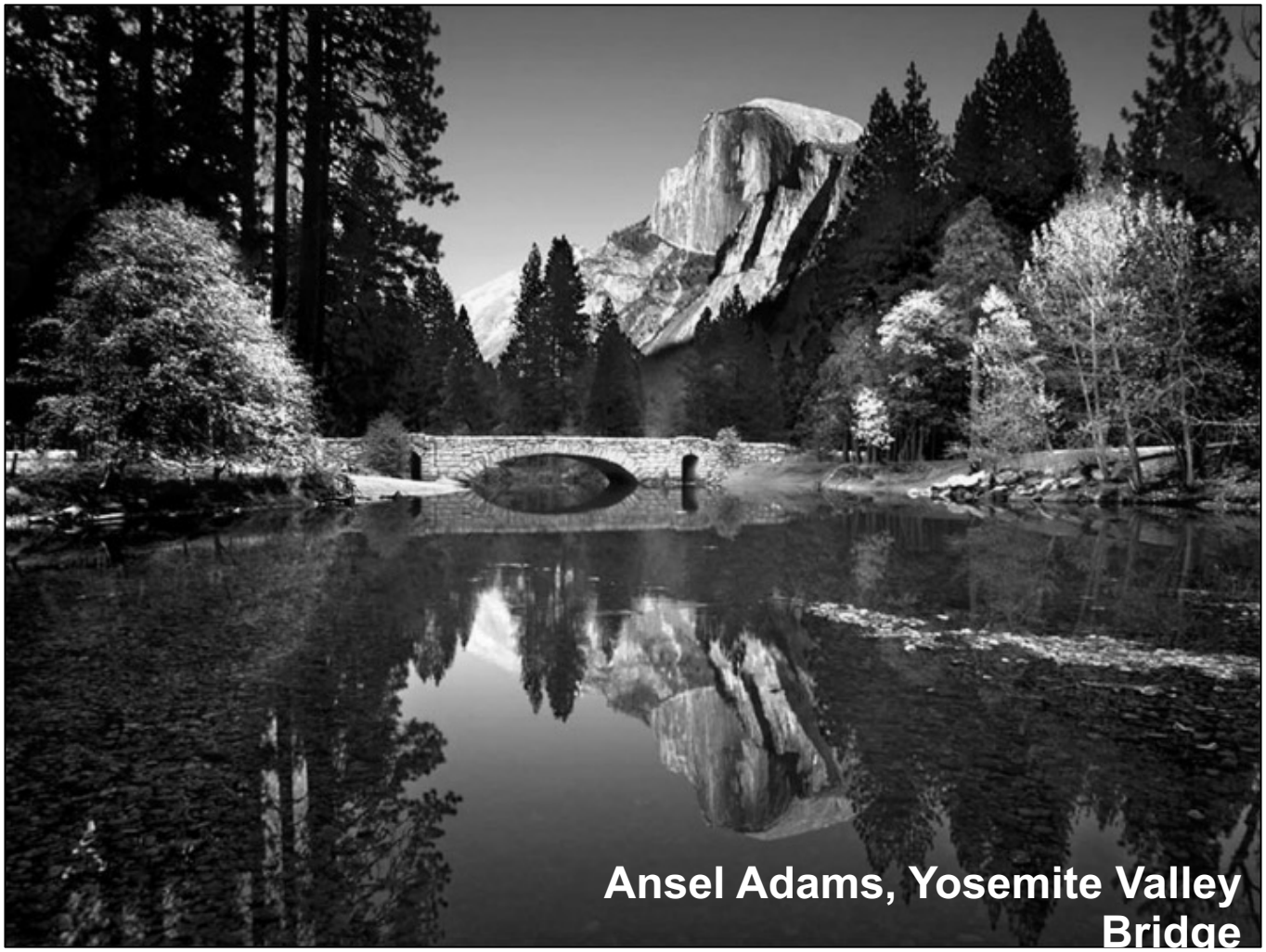
Pre-Training for R-CNN



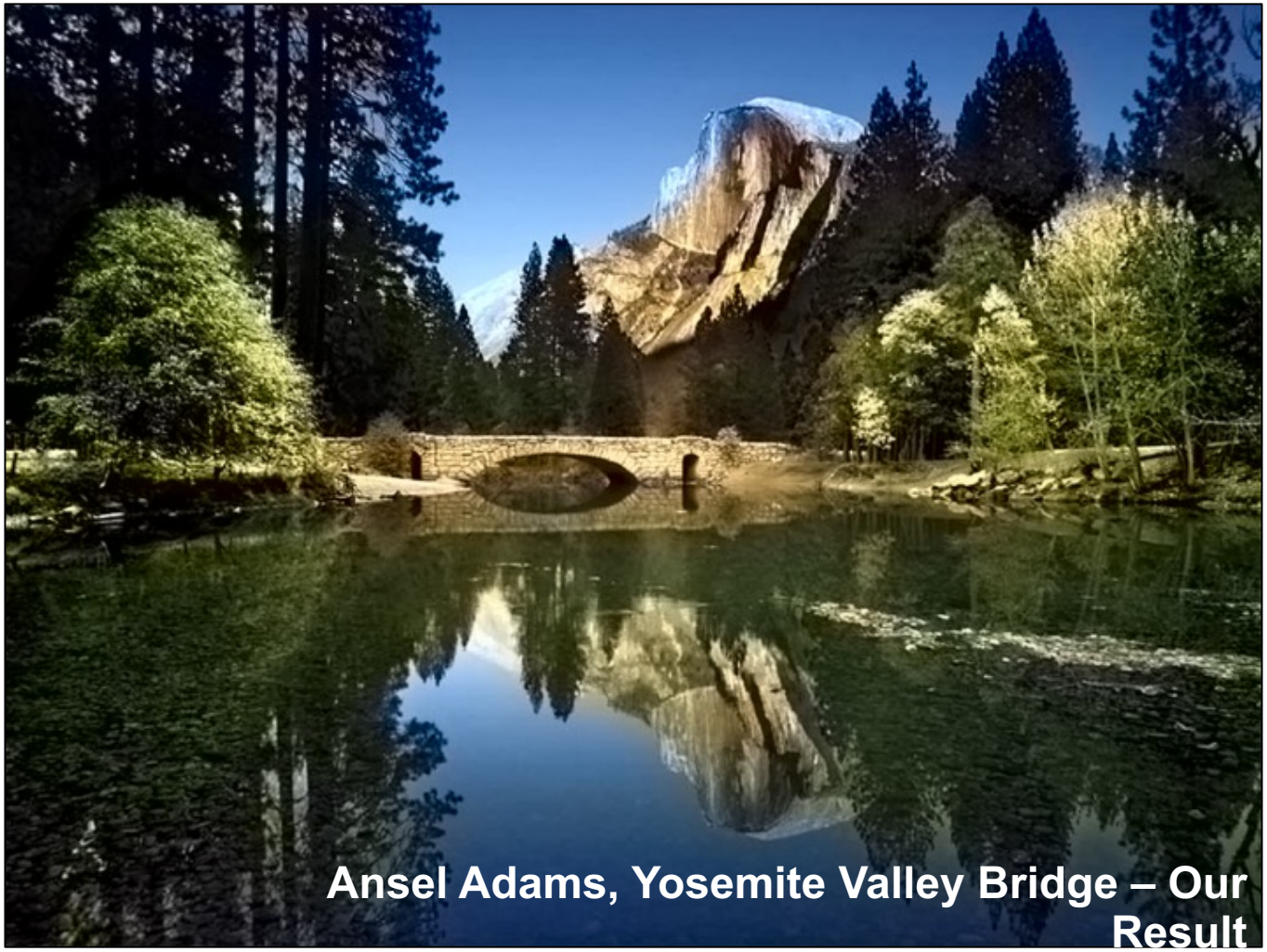
Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

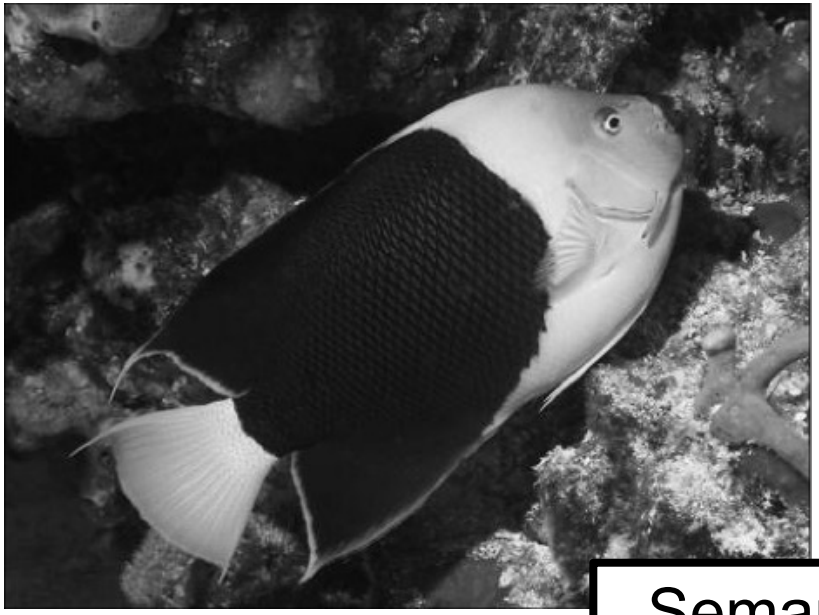
Other Sources Of Signal



Slide Credit: R. Zhang



Slide Credit: R. Zhang



$\mathcal{F}$

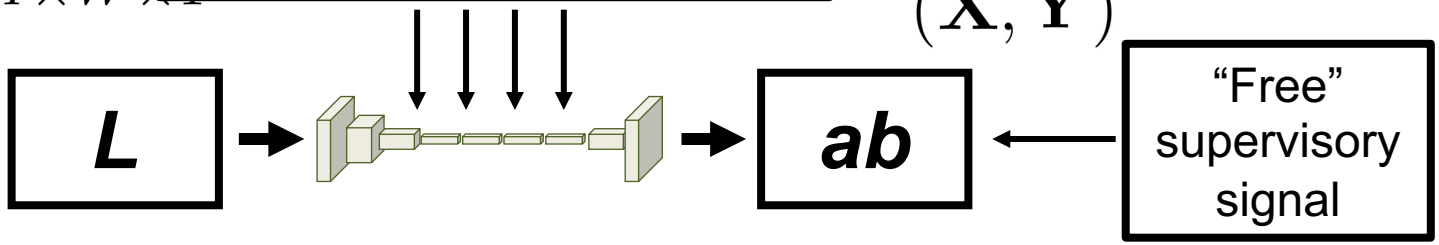
Grayscale image: *L*

$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Semantics? Higher-level abstraction?

oncatenate (L,ab)

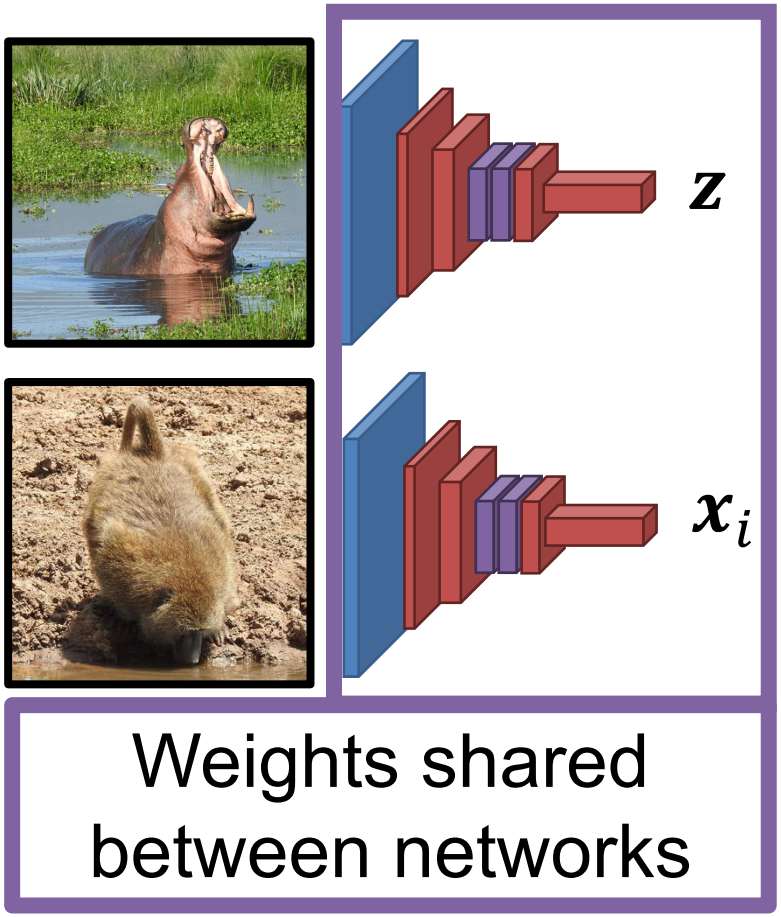
$(\mathbf{X}, \widehat{\mathbf{Y}})$



Slide Credit: R. Zhang


Visually Indicated Sounds
Andrew Owens
Phillip Isola
Josh McDermott
Antonio Torralba
Edward Adelson
William Freeman

Contrastive Learning



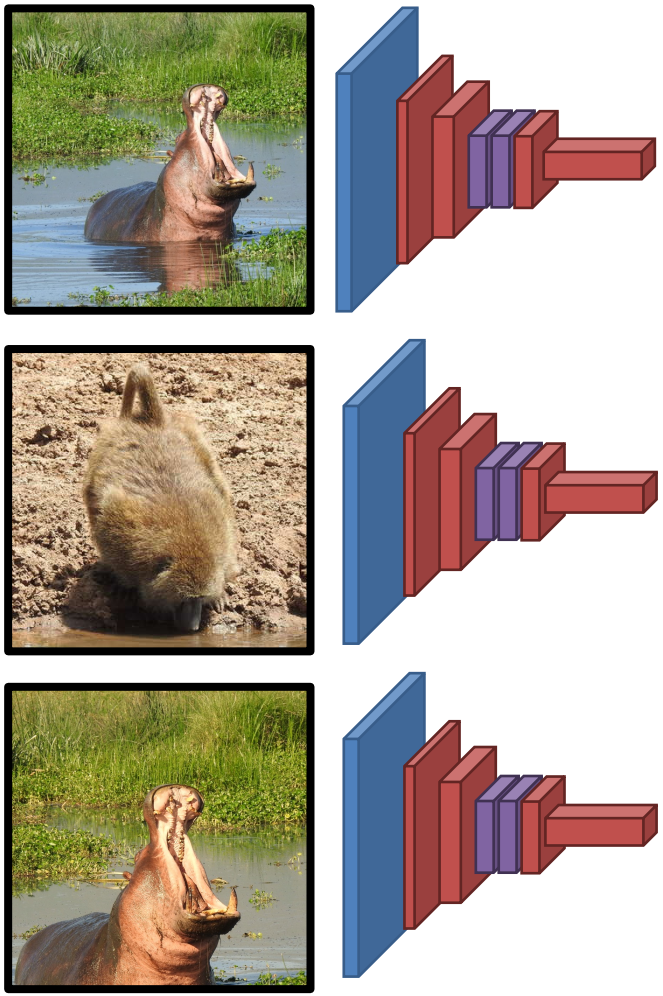
Given sample, construct feature z; take a bunch of other images, minimize:

$$-\log\left(\frac{\exp(\boldsymbol{z}^T\boldsymbol{z})}{\exp(\boldsymbol{z}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{z}^T\boldsymbol{x})}\right)$$

$$\frac{\exp(\boldsymbol{z}^T\boldsymbol{z})}{\exp(\boldsymbol{z}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{z}^T\boldsymbol{x})} \leq 1$$

He et al. Momentum Contrastive Learning, 2019. Wu et al. Instance discrimination 2018

Contrastive Learning



z

x_i

z'

Best performing methods measure distance to augmented sample z':

$$-\log\left(\frac{\exp(z^T z')}{\exp(z^T z') + \sum_i \exp(z^T x)}\right)$$

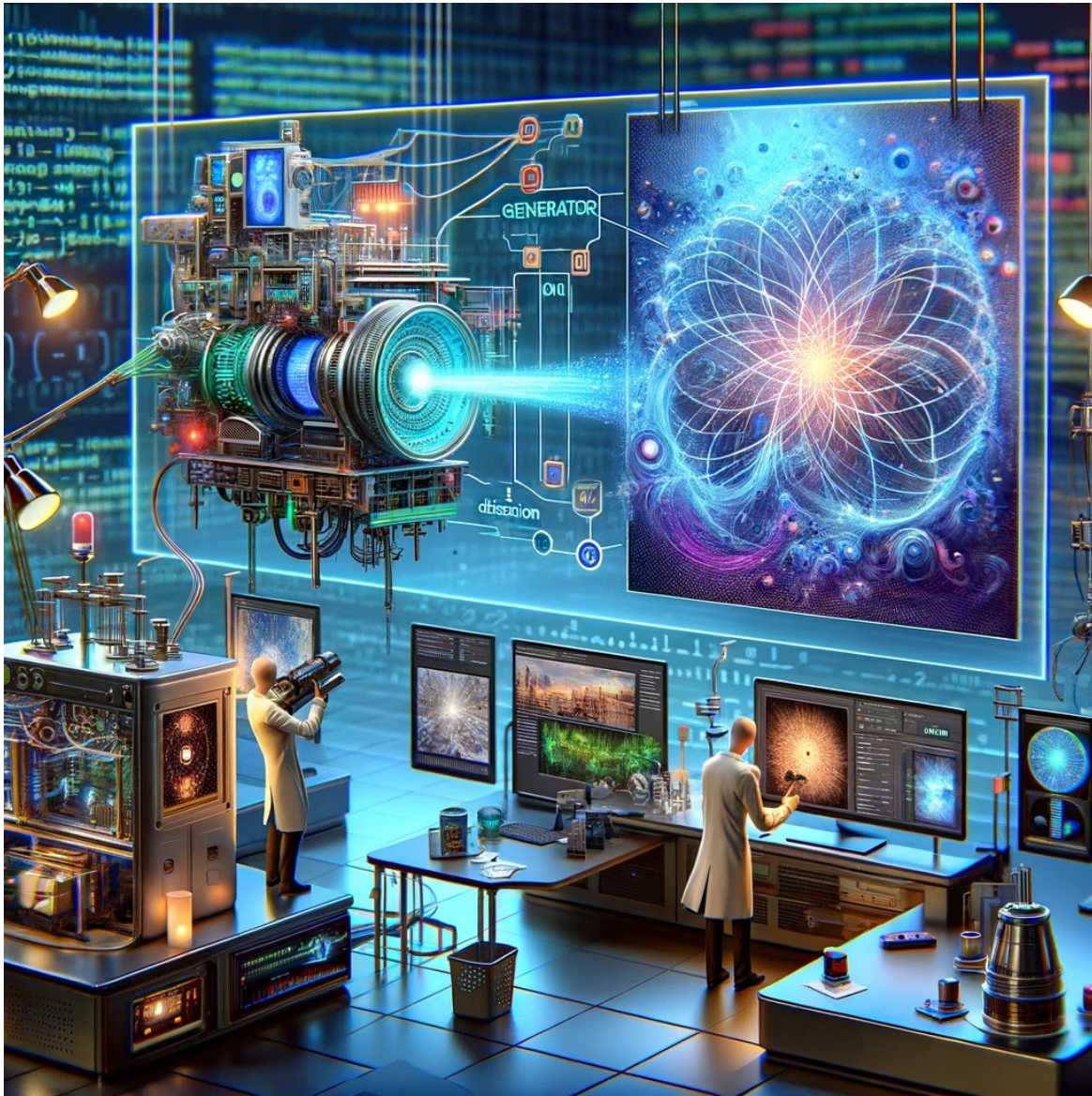
Goal: score augmented sample higher than everything else.

He et al. Momentum Contrastive Learning, 2019. Wu et al. Instance discrimination 2018

Visual Generative Models

- GANs
- Diffusion Models

→ By ChatGPT



Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Data: x

Label: y

Cat

Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Data: x

Label: y

Cat

Probability Recap:

Density Function
p(x) assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

$$\int_X p(x)dx = 1$$

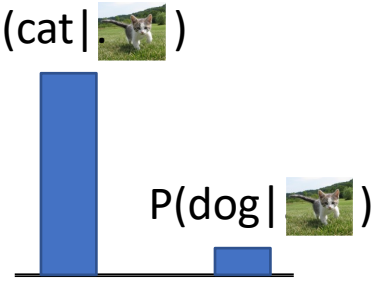
Different values of x **compete** for density

Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Data: x



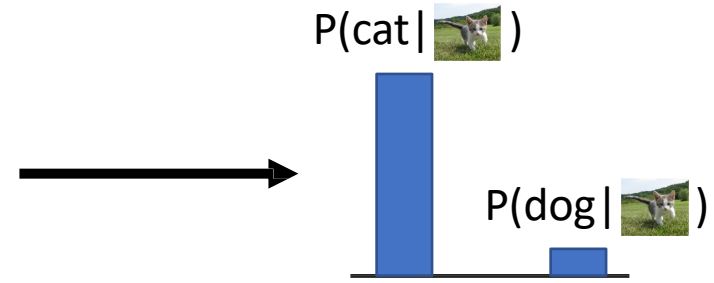
Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

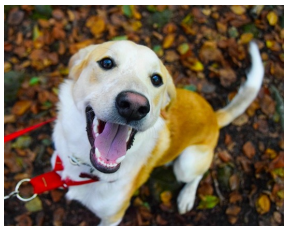
Slide Credit: Roni Sengupta

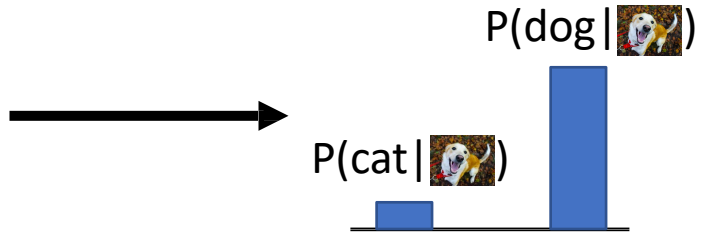
Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)



Generative Model: Learn a probability distribution p(x)





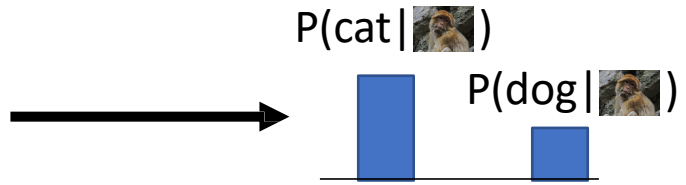
Conditional Generative Model: Learn p(x|y)

Discriminative model: the possible labels for each input "compete" for probability mass. But no competition between **images**

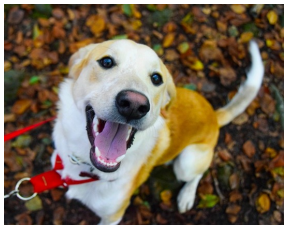
Slide Credit: Roni Sengupta

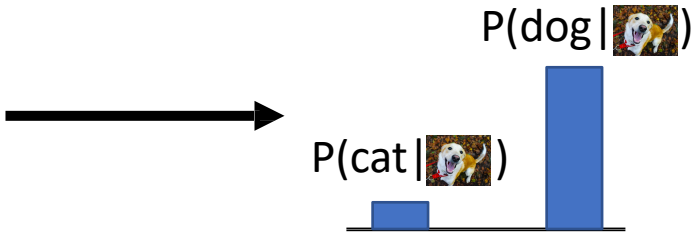
Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)



Generative Model: Learn a probability distribution p(x)





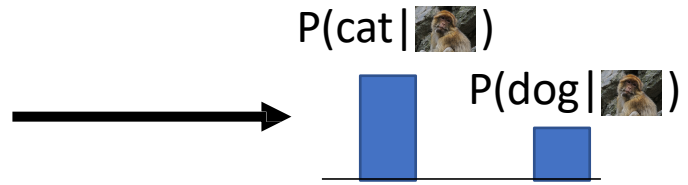
Conditional Generative Model: Learn p(x|y)

Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

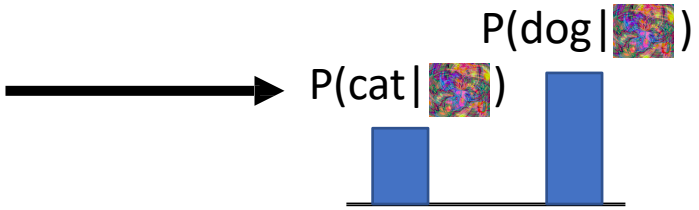
Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)



Generative Model: Learn a probability distribution p(x)



Conditional Generative Model: Learn p(x|y)

Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

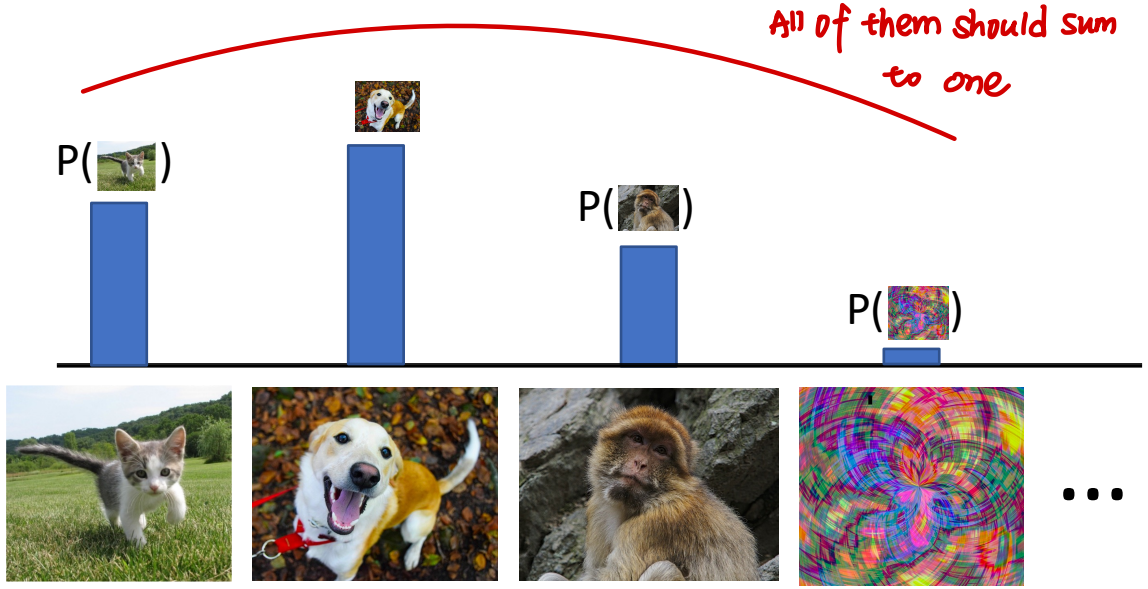
Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Generative model: All possible images compete with each other for probability mass (un likely).

Requires deep image understanding! Is a dog more likely to sit or stand? How about 3-legged dog vs 3-armed monkey?

Model can "reject" unreasonable inputs by assigning them small values

Slide Credit: Roni Sengupta

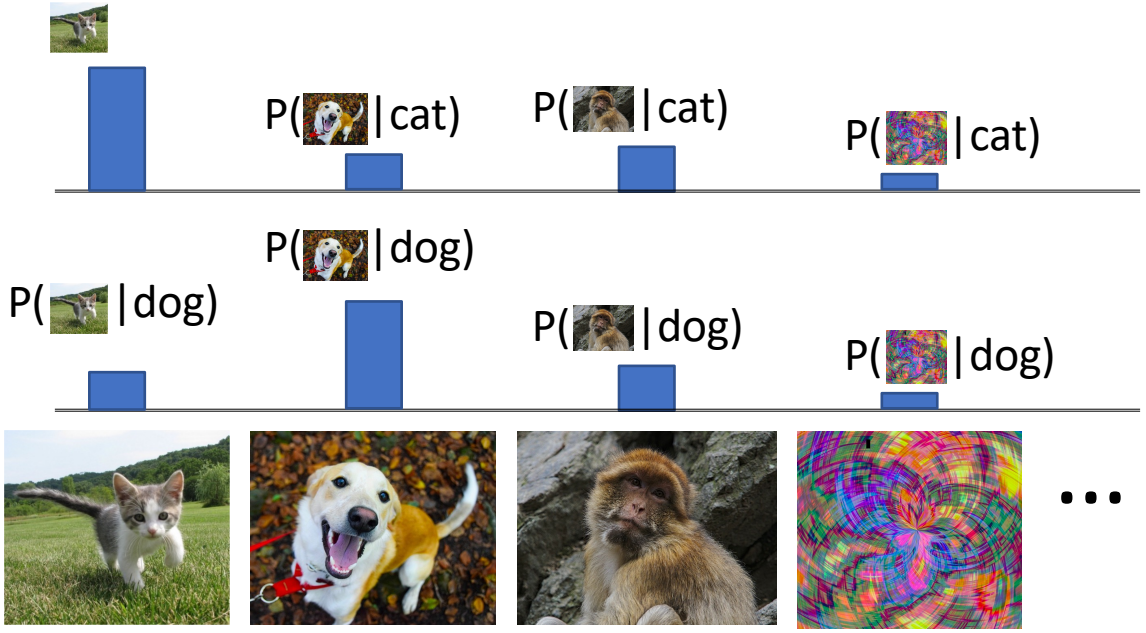
Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

given the label.



Conditional Generative Model: Each possible label induces a competition among all images

Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Recall **Bayes' Rule:**

$$P(x \mid y) = \frac{P(y \mid x)}{P(y)} P(x)$$

Slide Credit: Roni Sengupta

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Recall **Bayes' Rule:**

$$P(x \mid y) = \frac{P(y \mid x)}{P(y)} P(x)$$

Conditional Generative Model

Discriminative Model

Prior over labels

(Unconditional) Generative Model

your intuition or some statistic of the world say that there are more dogs' labels than cats' labels. (prior distribution of the world)

We can build a conditional generative model from other components, etc.

Slide Credit: Roni Sengupta

Progress in Generative Models of Images

Slide credit: Ian Goodfellow, 2019

https://github.com/NVlabs/stylegan3 [Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

Images and Text

TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES

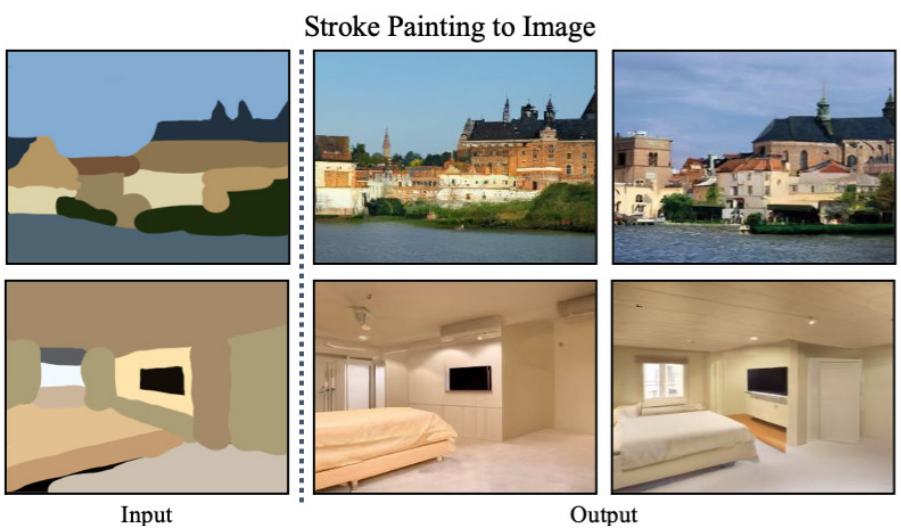
Edit prompt or view more images↓

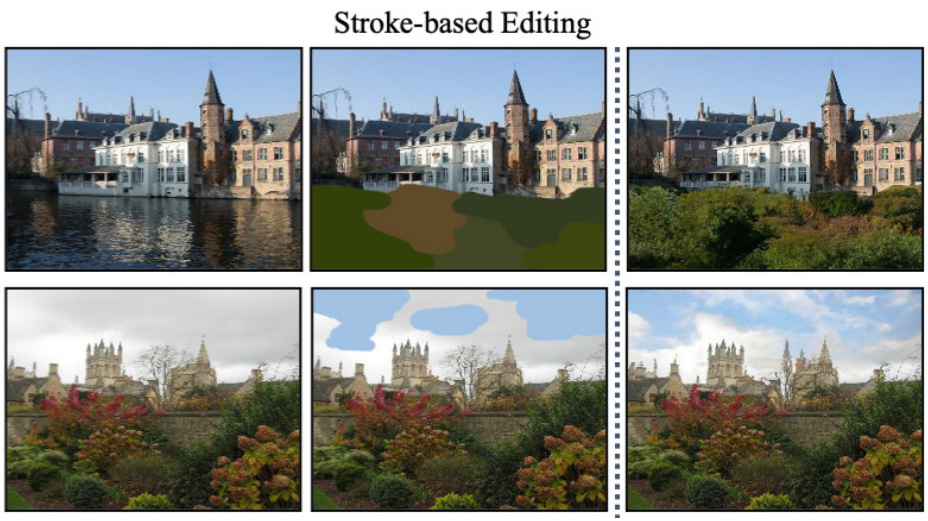
P(image | caption)

TEXT PROMPT a store front that has the word ‘openai’ written on it. . . .

AI-GENERATED IMAGES

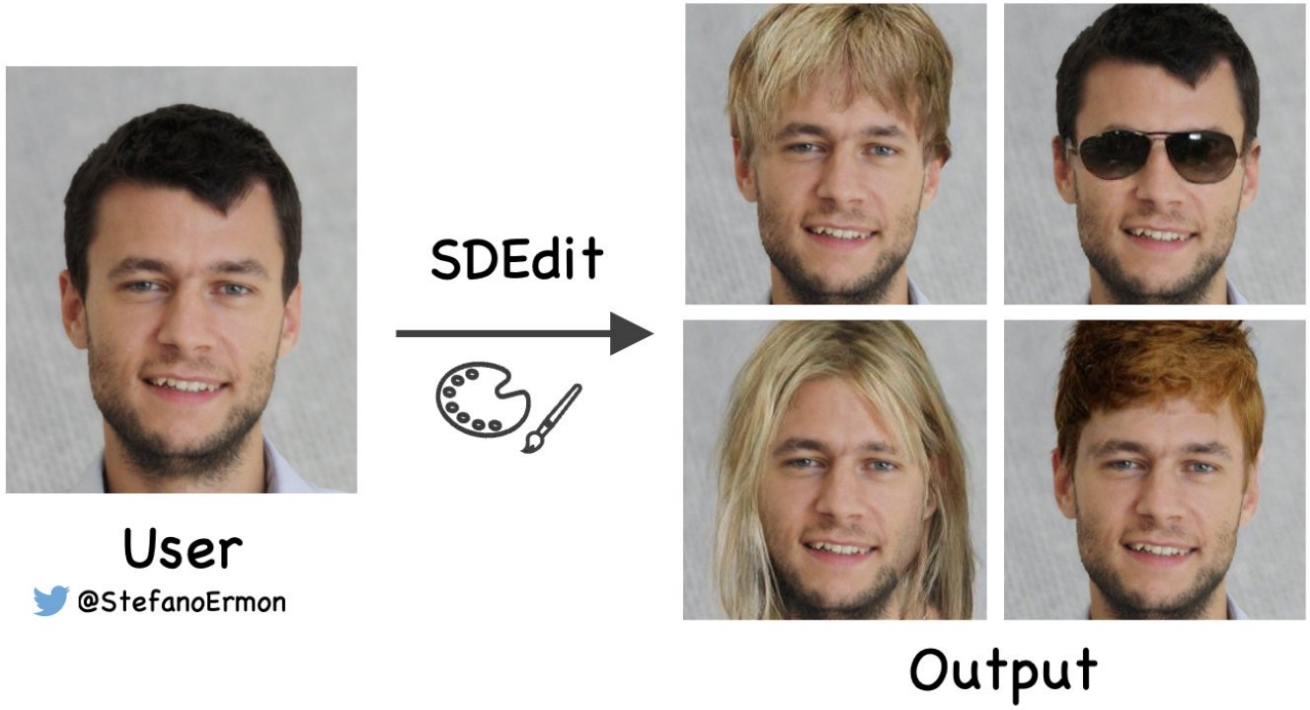
Inverse Problems with GenAI





Inverse Problems with GenAI

Which image is real?

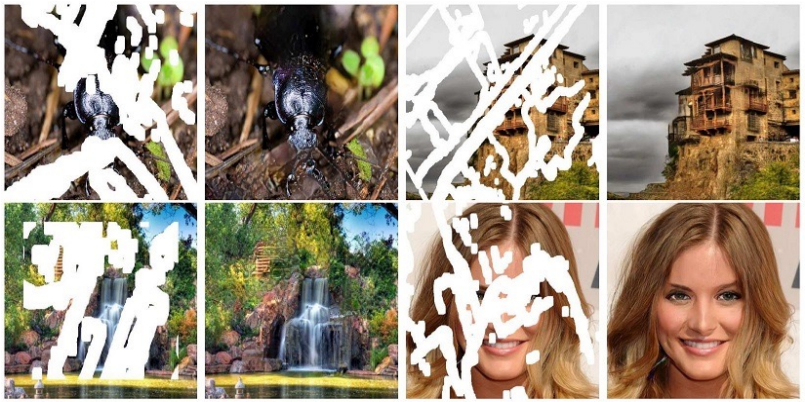


Inverse Problems with GenAI

P(high resolution | low resolution)

Menon et al, 2020

P(full image| mask)



Liu al, 2018

P(color image| greyscale)

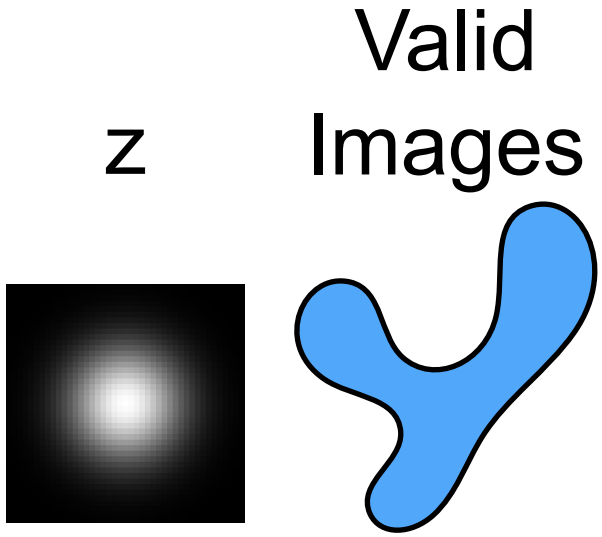
Antic, 2020

How Many Images Are There?

- Set height and width to 1024
- Assume 256^3 (aka 2^{24}) values per pixel
- **How many images can I create?**
- (256^3)~1M
- **Why might it be quite a bit less?**

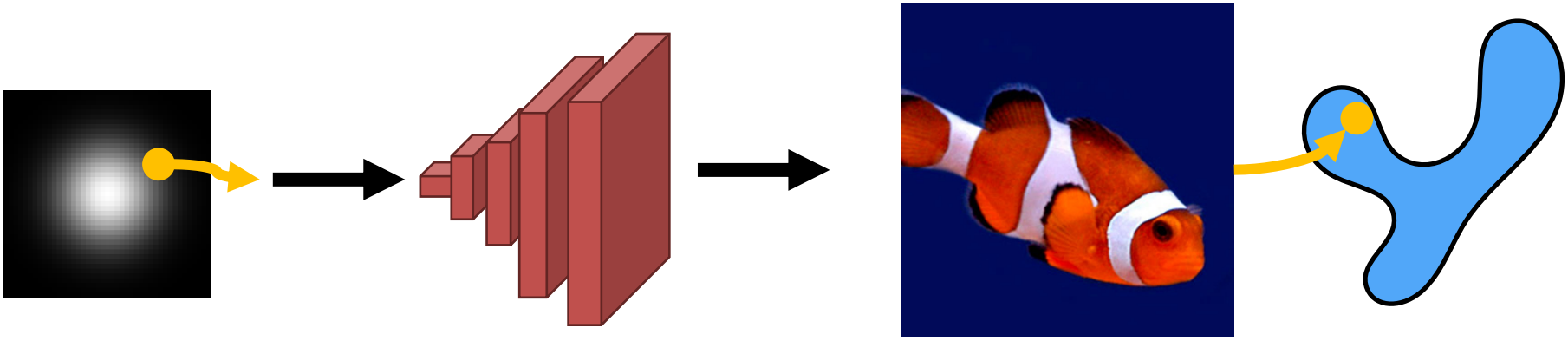
Learning a Mapping

- Want to learn a mapping
- **From:** a “latent” space z, often assume to be the result of sampling N-D Gaussian noise
- **To:** the space of valid images



Generating Data

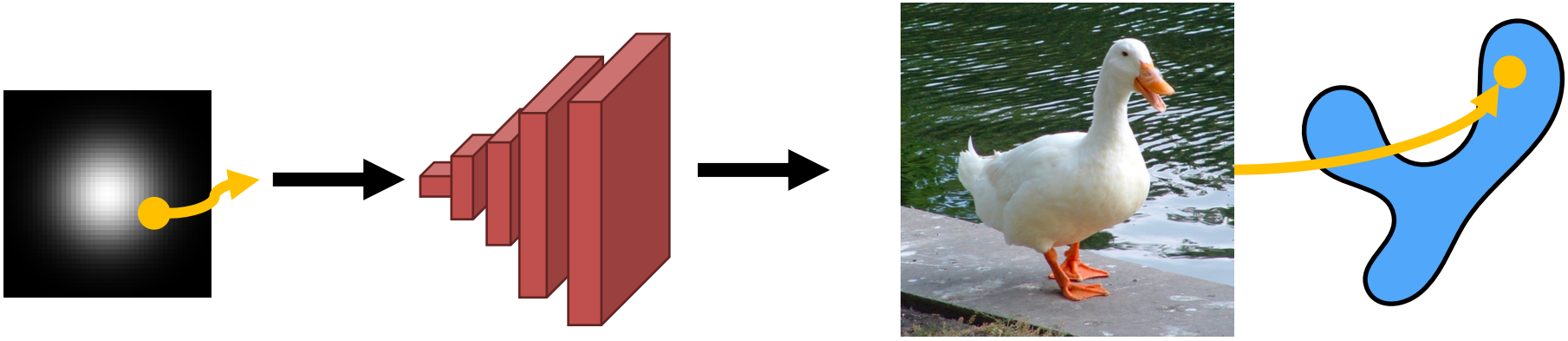
z G(·) x Valid Images



Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generating Data

z G(·) x Valid Images



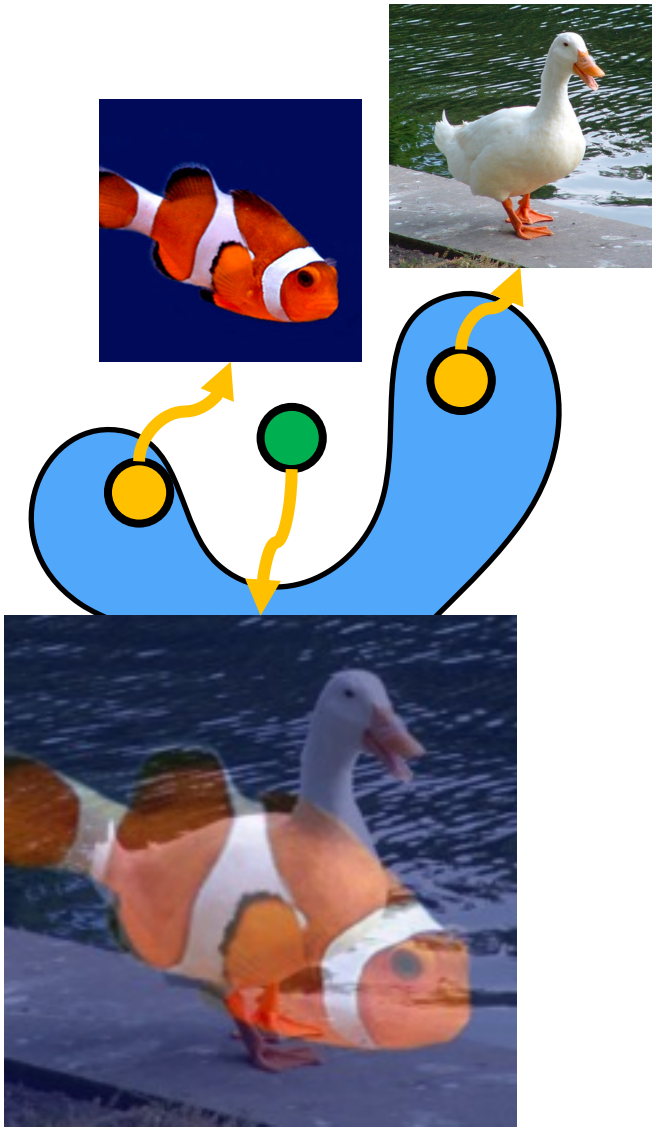
Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Why The Funny Shape?

Given two valid images, what about their average?

Key things to remember

- Linear combinations of images aren't valid images. (off-manifold)
- Explains funny shape ("manifold") and need for a deep network



Generative Adversarial Networks

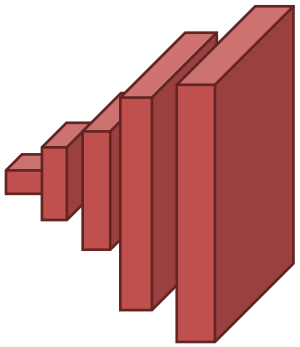
- Generator tries to make fake images – accepts noise and makes an image
- Discriminator tries to identify fakes – outputs p(~~fake~~ real)

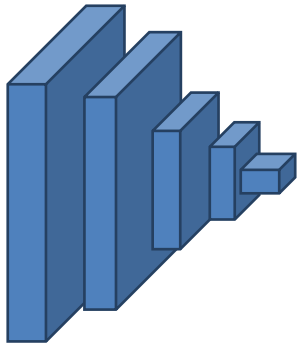
feed both generated image & real image

Generator

Discriminator

z G(·) G(z) D(·)



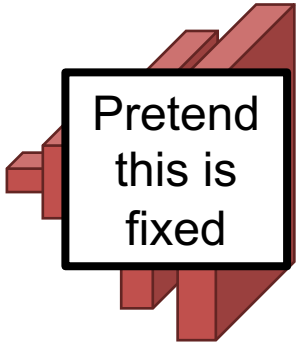


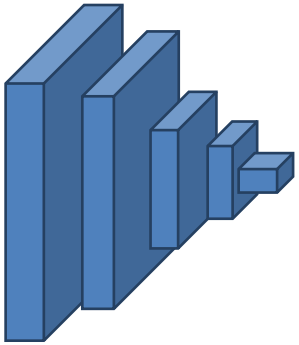
Real vs fake

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generative Adversarial Networks

z G(·) G(z) D(·)

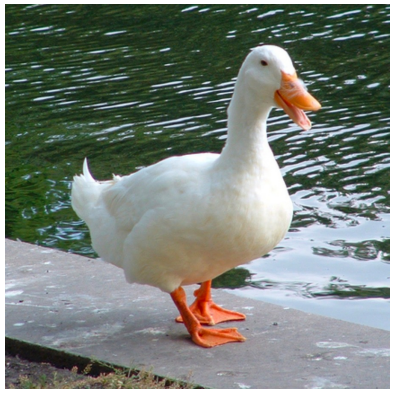


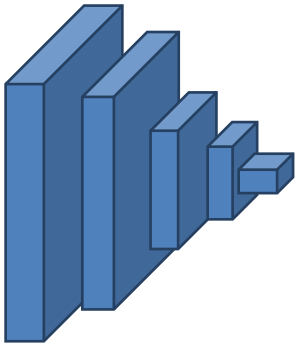


Want: fake (low)

D(x) = p(x is real)
Want to maximize log p(real) for real data

x D(·)





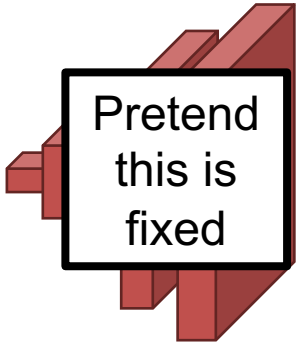
Want: real (high)

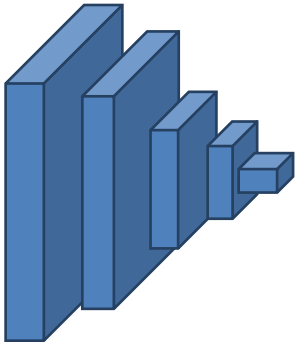
$$\arg\max_{D} E_{z,x}[\log\left(1 - D\left(G(z)\right)\right) + \log\left(D(x)\right)]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generative Adversarial Networks

z G(·) G(z) D(·)





Want: fake (low)

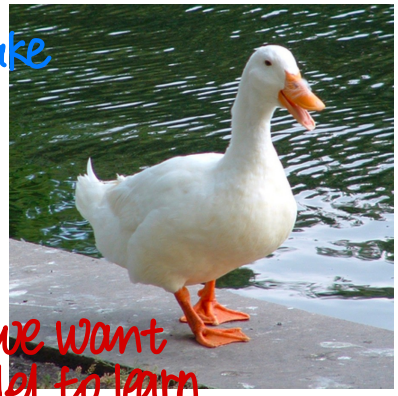
Gz is real

1-D(G(z)) = Gz is fake

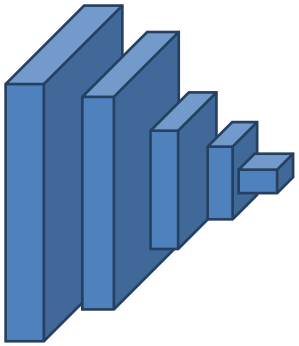
p(G(z) is fake)

Want to maximize log p(fake) for fake data

x D(·)



Because we want the model to learn.

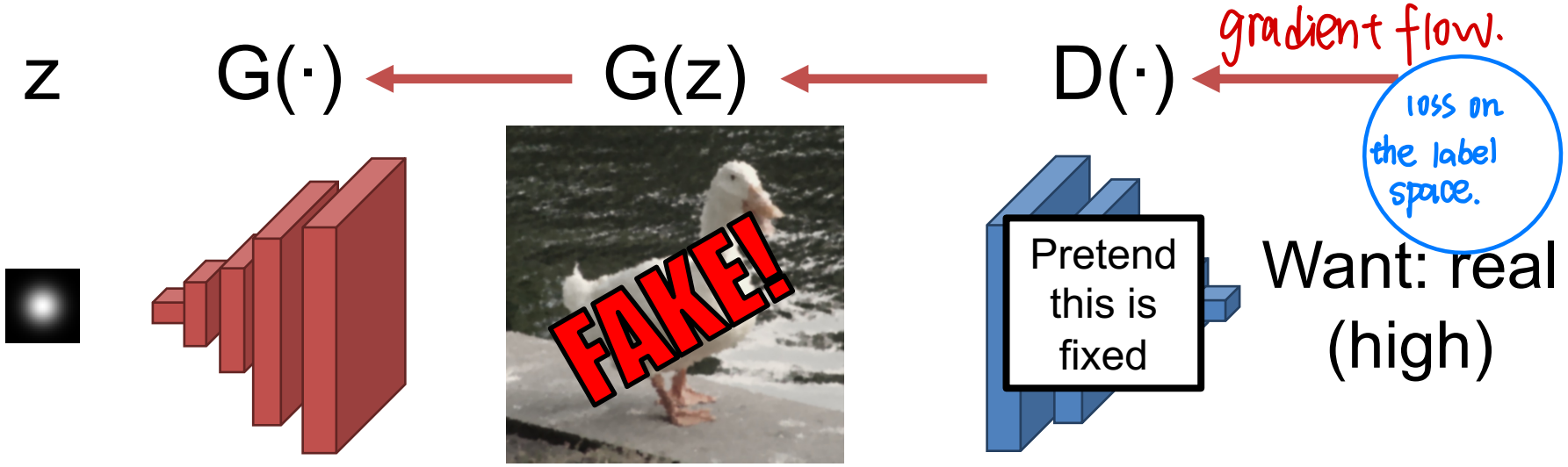


Want: real (high)

$$\arg\max_{D} E_{z,x}\left[\log\left(1 - D\left(G(z)\right)\right) + \log\left(D(x)\right)\right]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generative Adversarial Networks



Goal of generator G: fool the discriminator D while getting to use gradients from D

Basically what you do: Analyzing the features learned by D by classifying this real and fake. and then using these features learned by D to supervise signals to the generator.

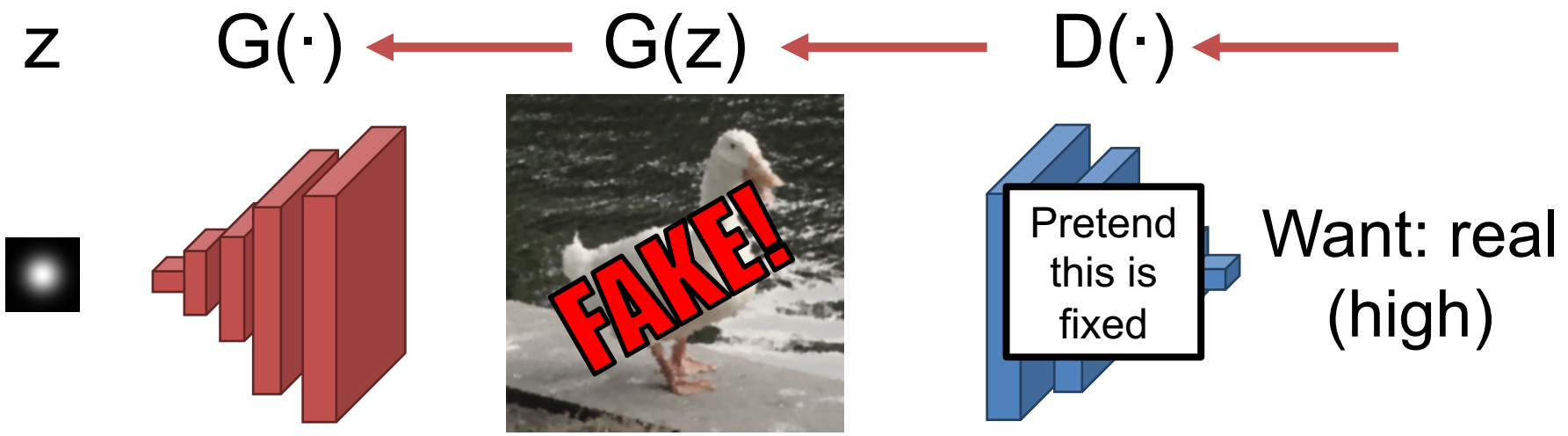
$$\arg\min_G E_z[\log\left(1 - D\left(G(z)\right)\right)]$$

1-D(G(z)) = p(G(z) is fake)

Want to ~~maximize~~ minimize log p(fake) for fake data

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generative Adversarial Networks



Goal of generator G: fool the discriminator D while getting to use gradients from D

Intuition: make D think G(z) is real

Analogy: art forger and art detective

try to fake an image that looks like the original.

try to find features that marks the different

How good are you at spotting forgeries?

$$\arg\min_G E_z[\log\left(1 - D\left(G(z)\right)\right)]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Generative Adversarial Networks

Final goal: find the generator that fools the best D that you could find.

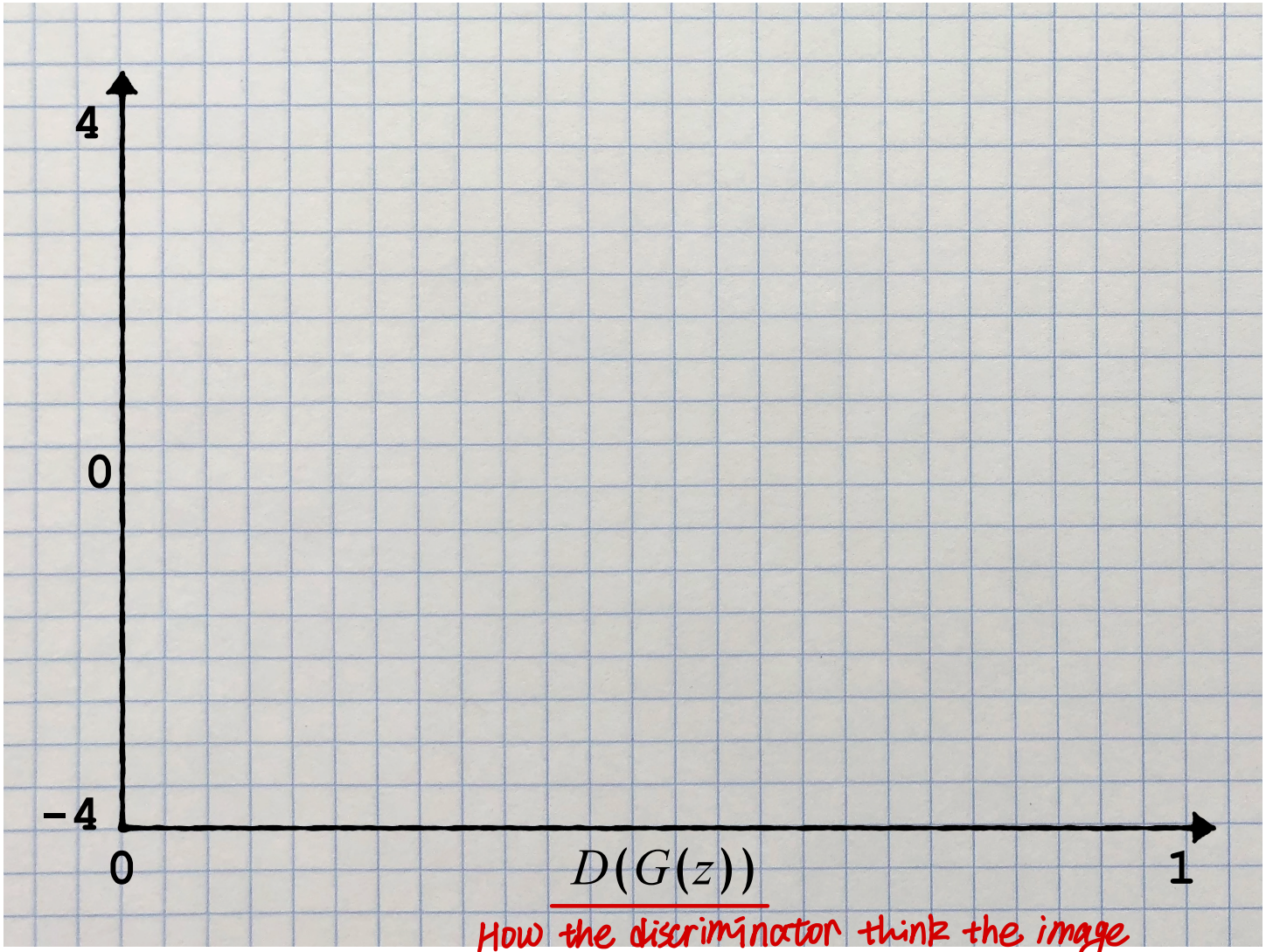
Min-max game between G and D

In practice, important not to let the discriminator get too good. **Why?**

$$\arg\min_{G}\max_{\mathrm{D}} E_{z,x}[\log\left(1-D\left(G(z)\right)\right)+\log\left(D(x)\right)]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

Caveat on G Loss

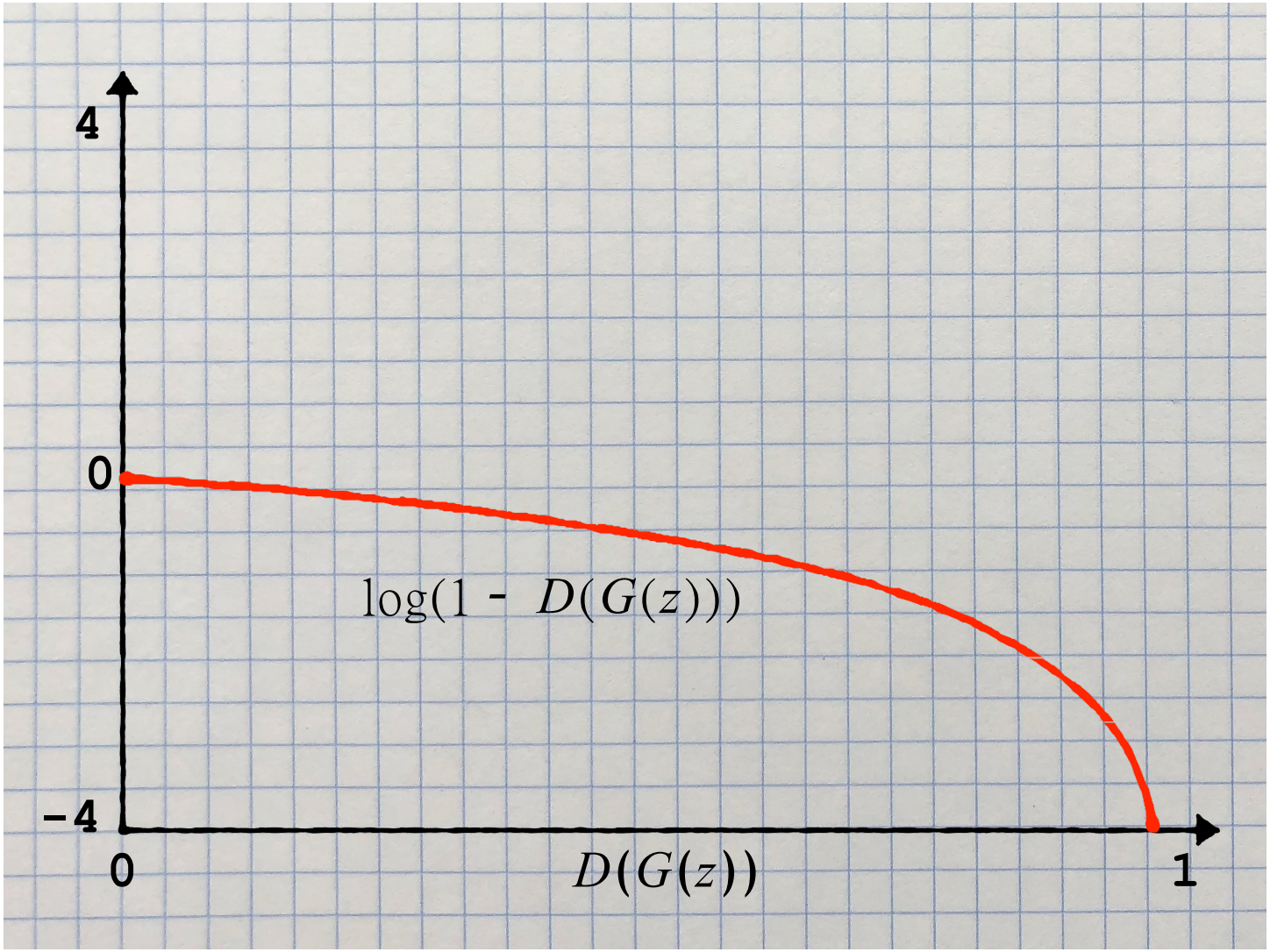


How the discriminator think the image created by the generator. (fack ⟷ real)

0 1

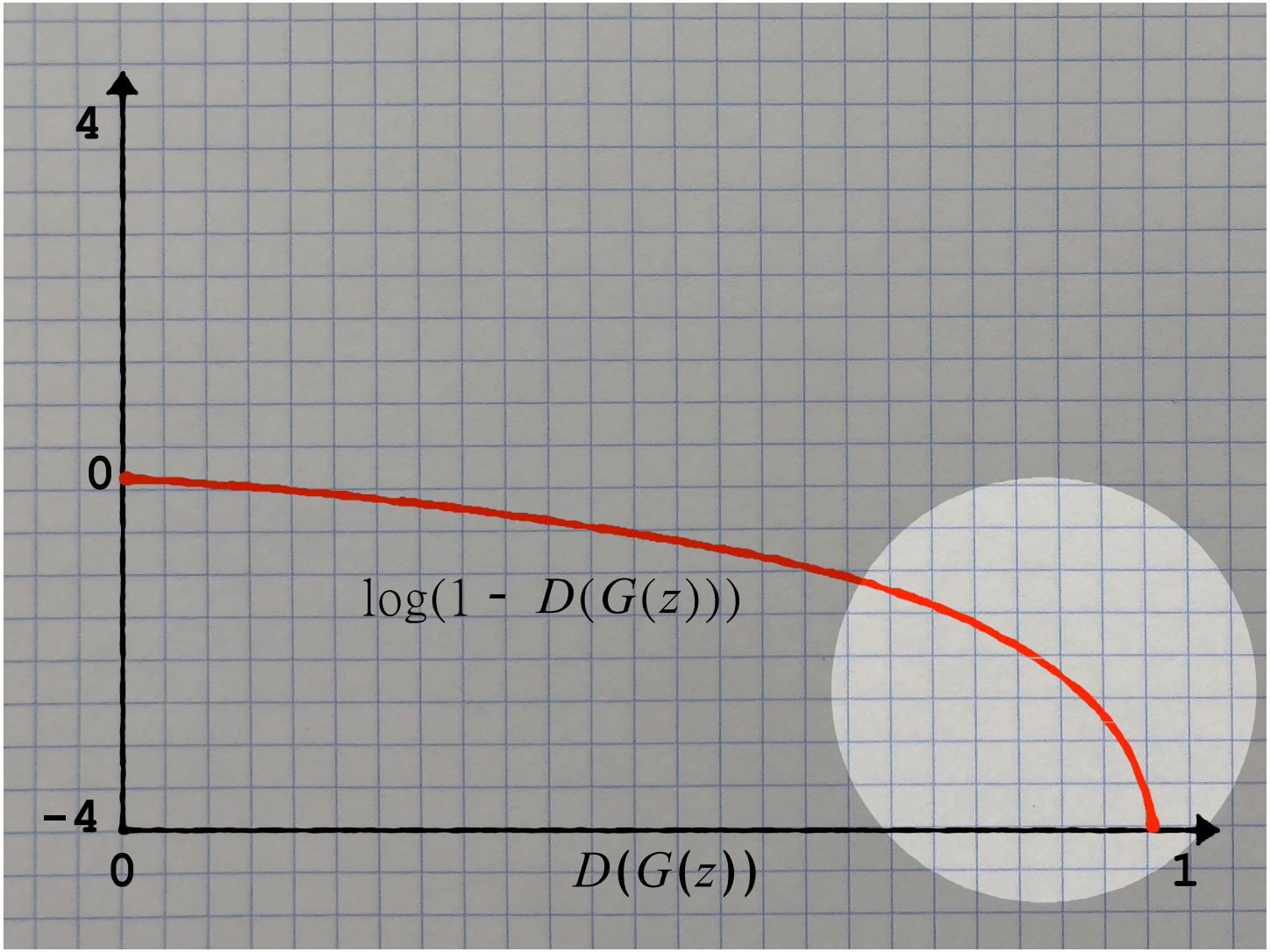
Slide Credit: Roni Sengupta

$$\arg\min_G E_z[\log\left(1 - D\left(G(z)\right)\right)]$$

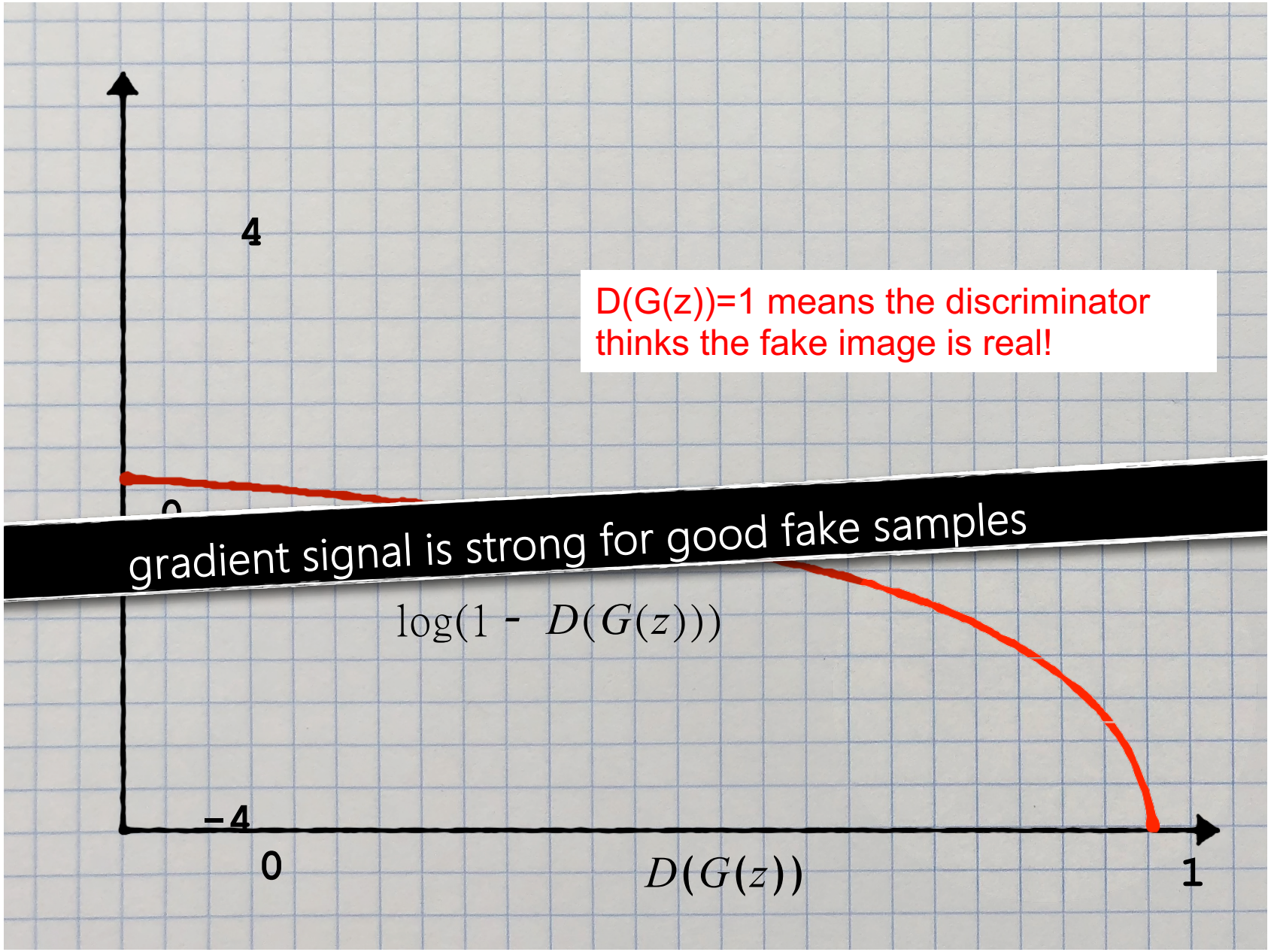


Slide Credit: Roni Sengupta

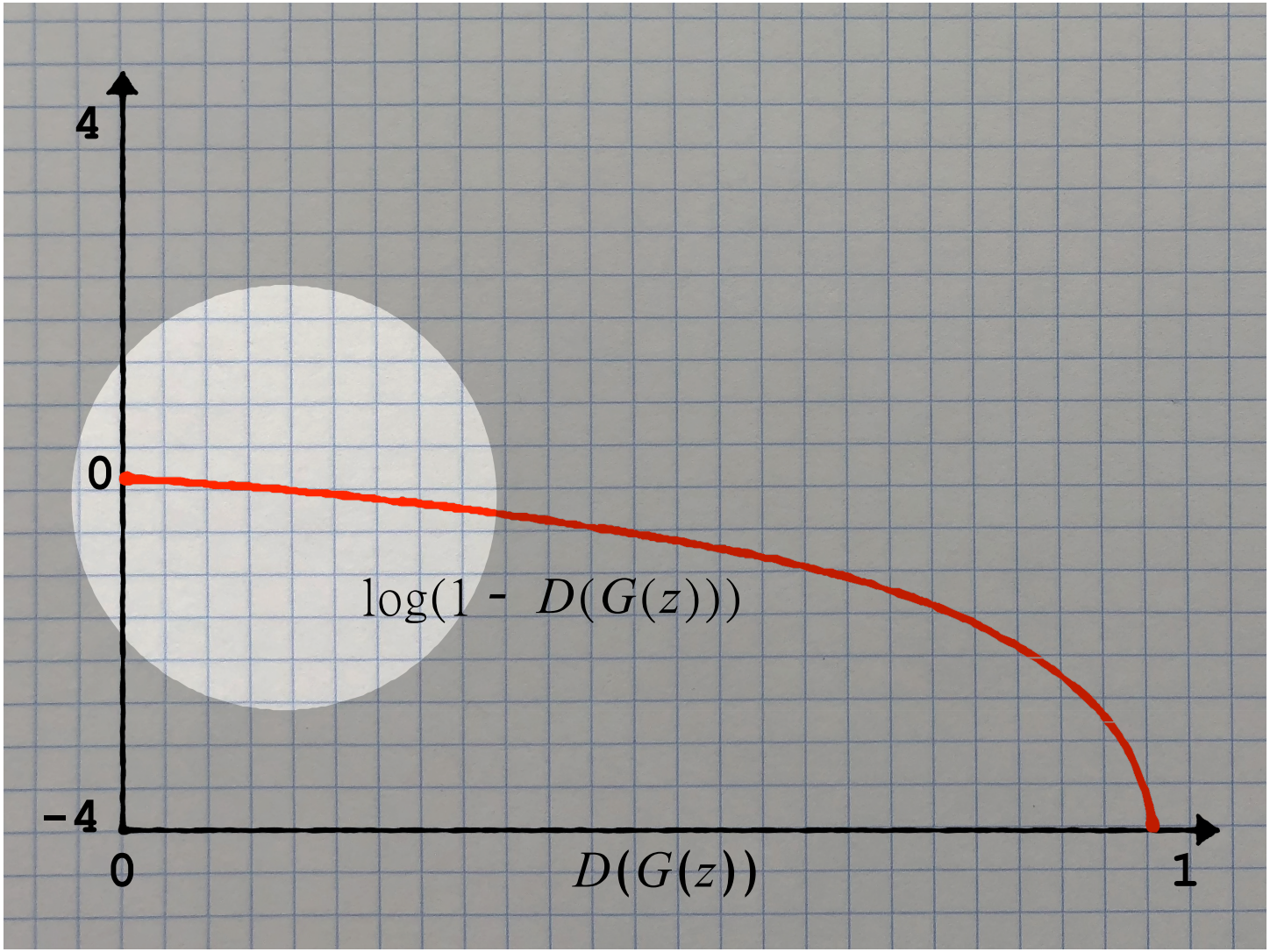
$$\arg\min_G E_z[\log\left(1 - D\left(G(z)\right)\right)]$$



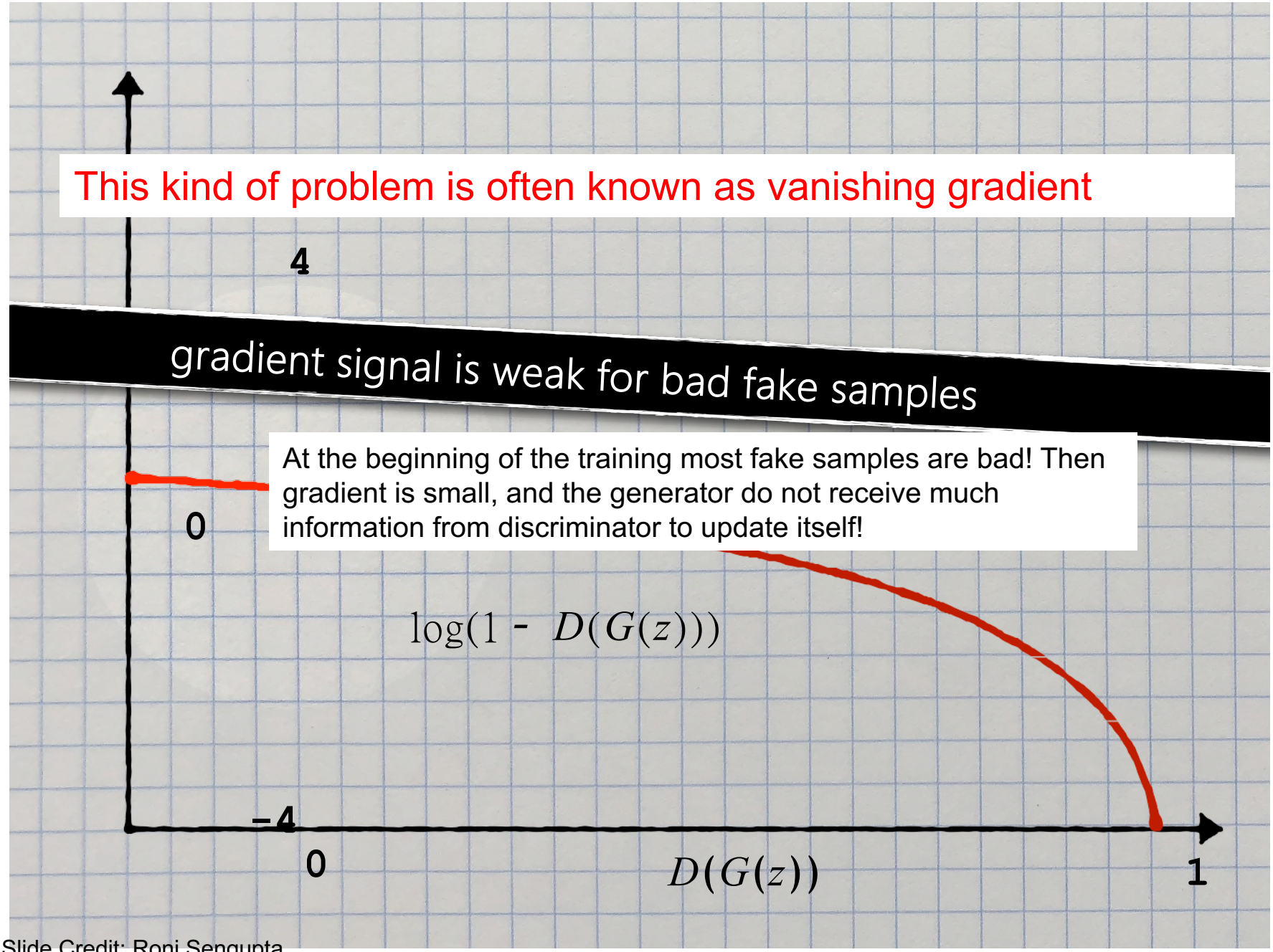
Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta

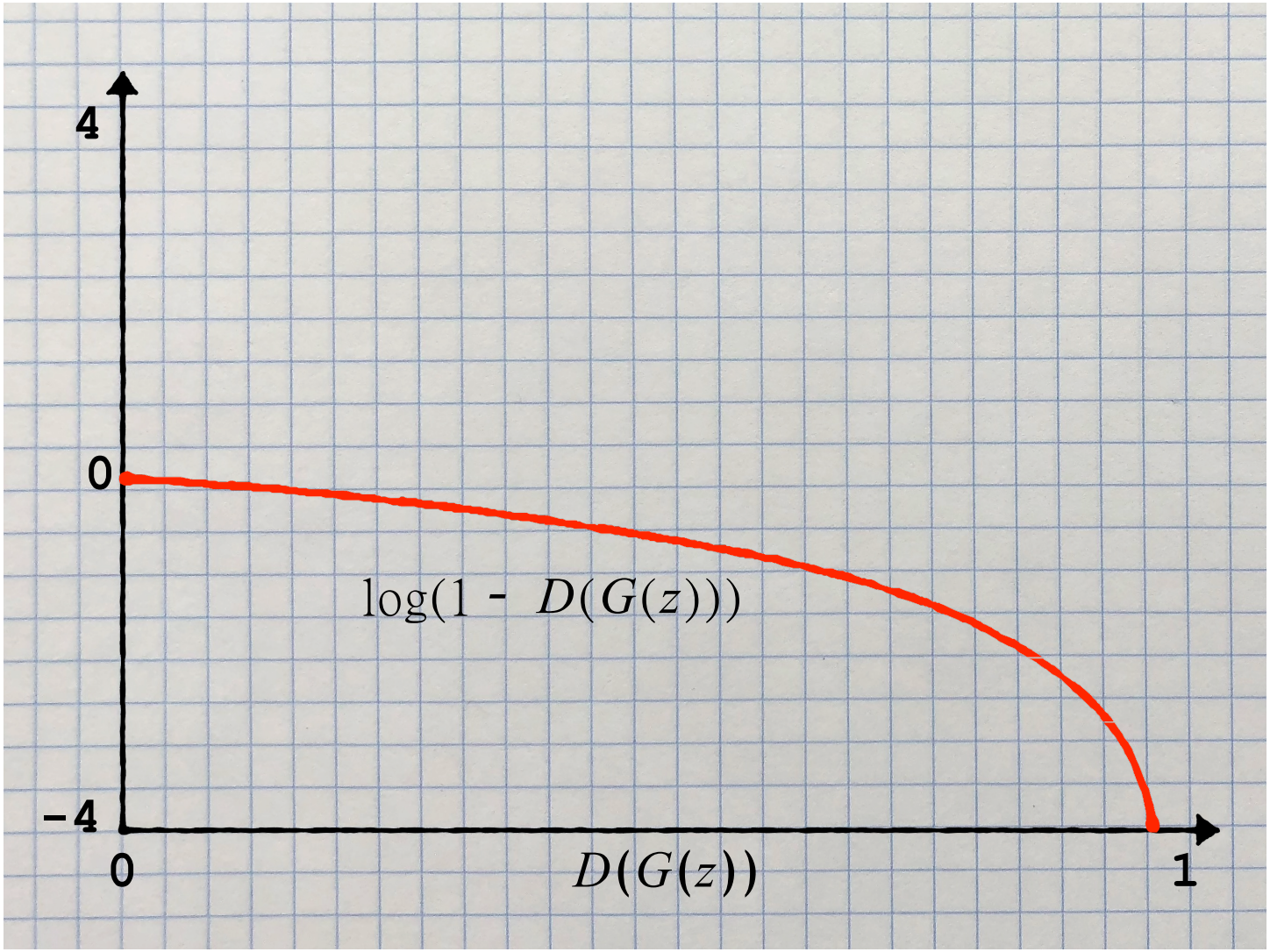
Improved G Loss

$$E_z[\log\left(1 - D\left(G(z)\right)\right)$$

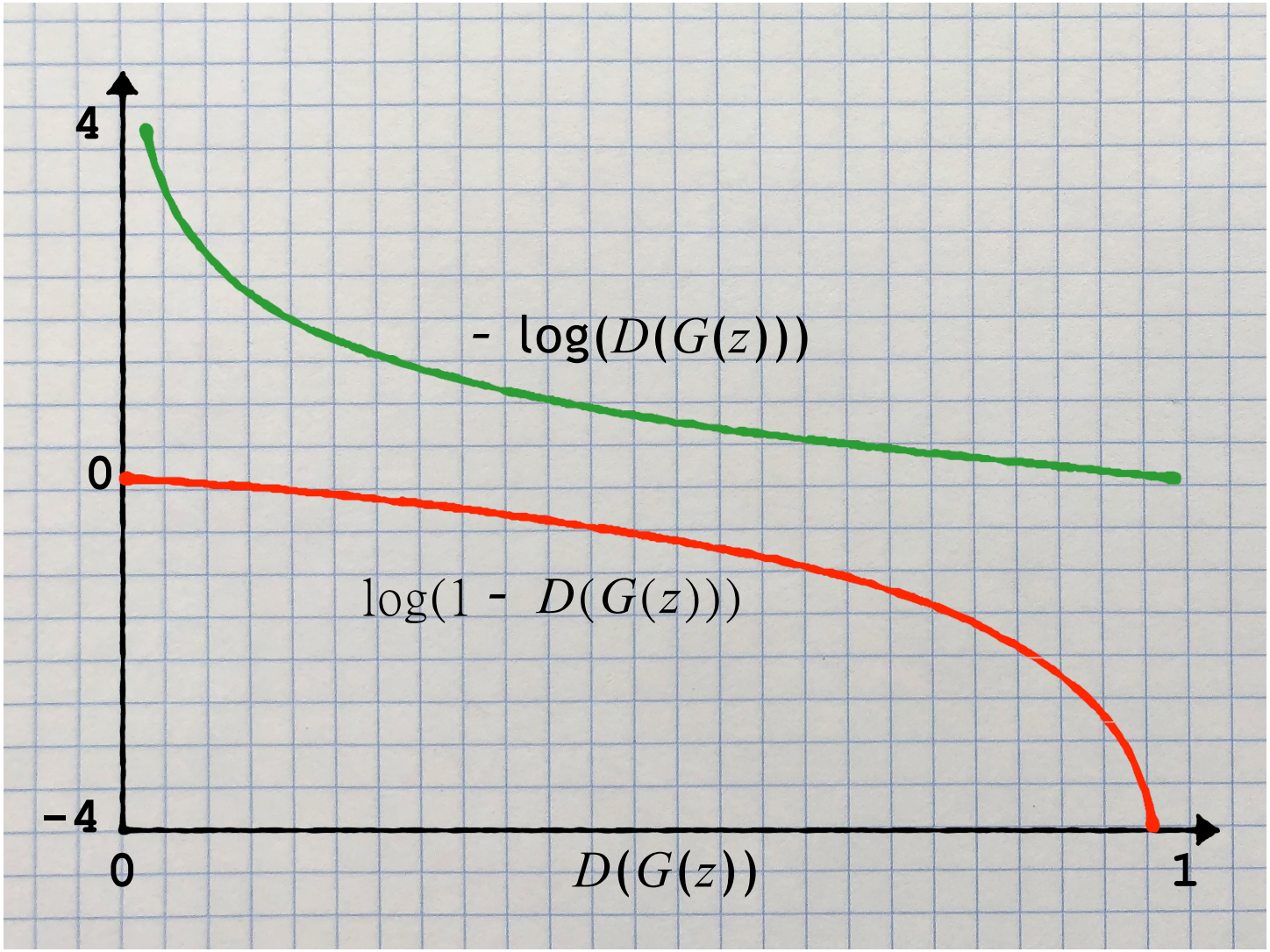
Replace with

$$E_z[-\log\left(D\left(G(z)\right)\right)$$

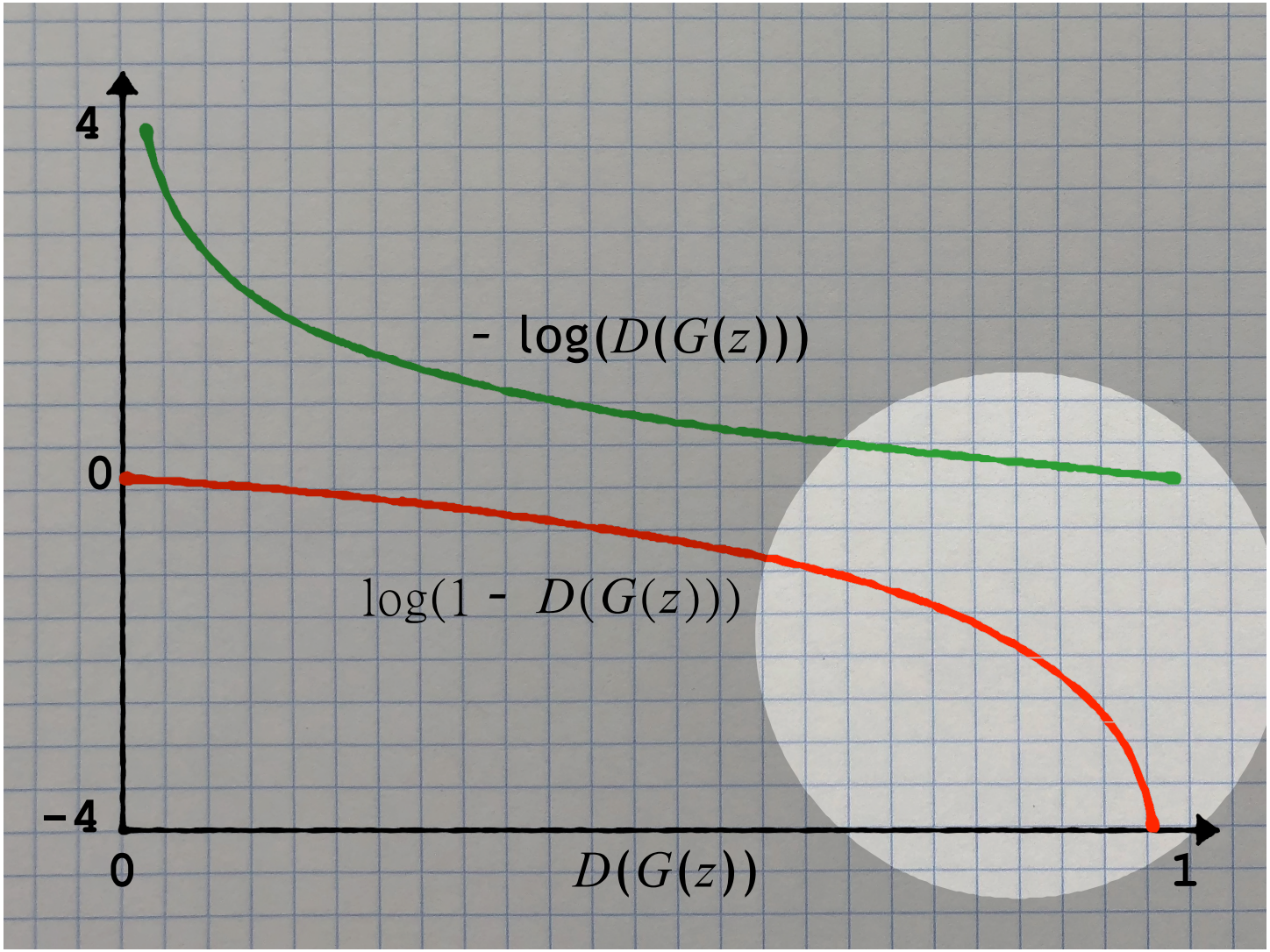
Slide Credit: Roni Sengupta



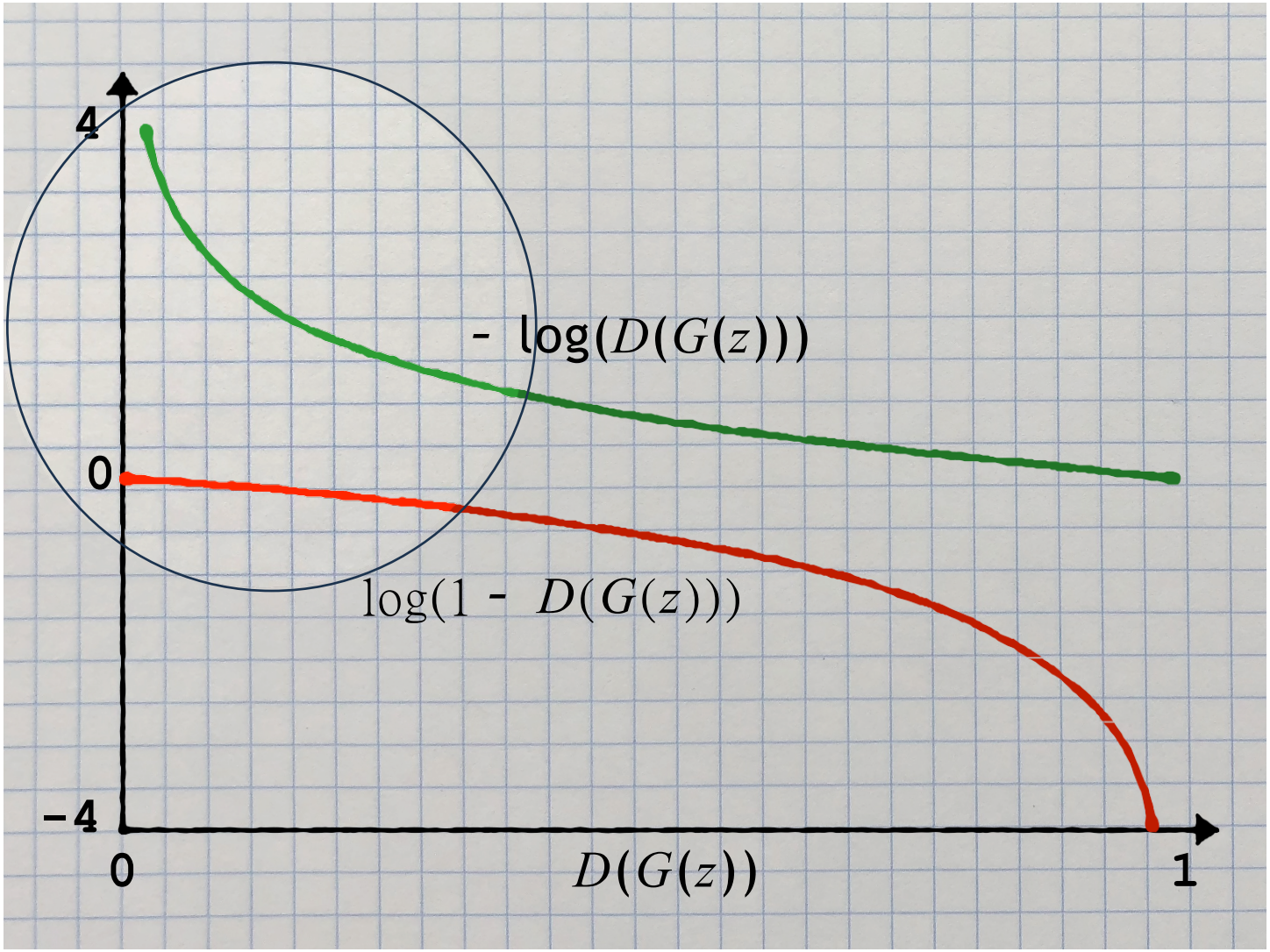
Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta



Slide Credit: Roni Sengupta

Typically for GAN Training

- Very often unstable since it's complex
- Need regularizations or more stable losses (out of scope) to make the discriminator weaker.
- Need balance between D and G
- Training discriminator is easier than generator (more people can appreciate art vs # of artists)

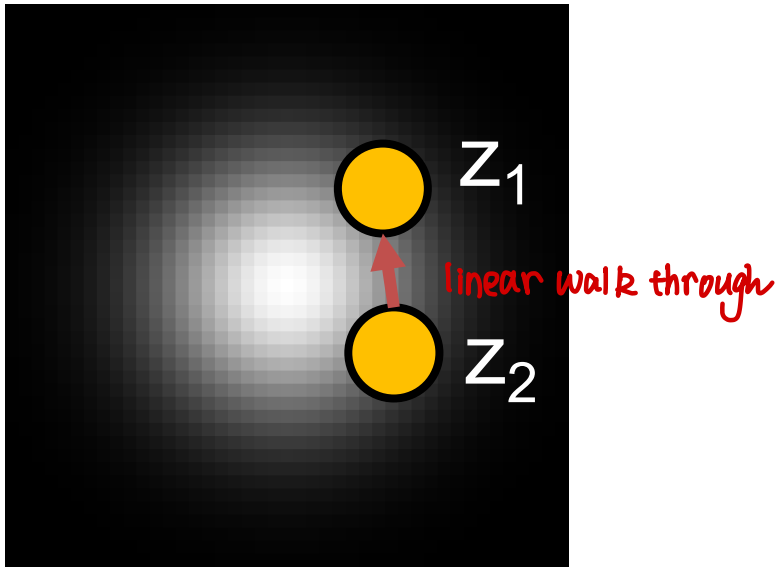
GAN Summary

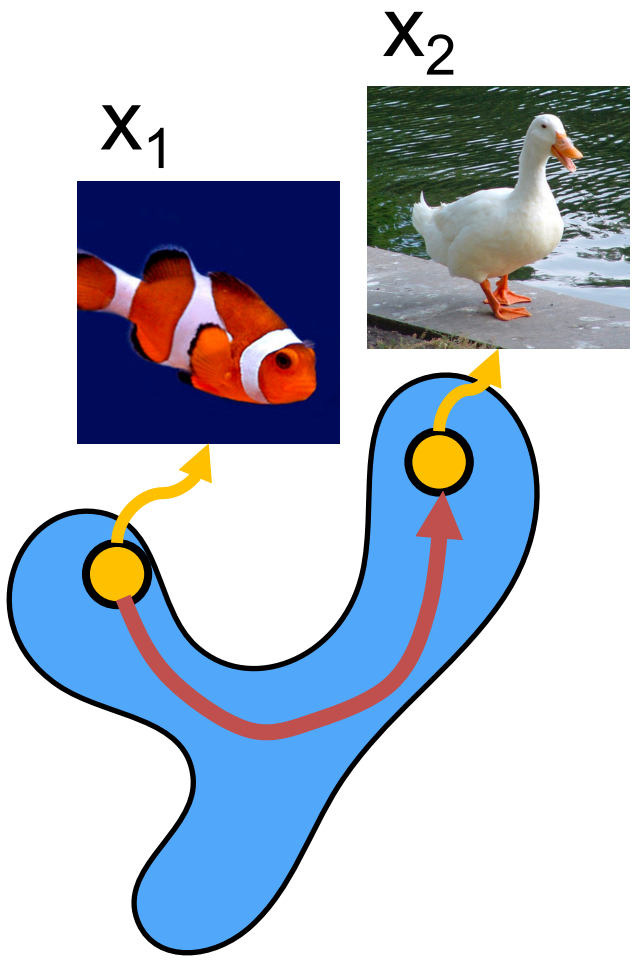
```
# Set batch size
for number of training iterations do
    # D update
    Sample noise vectors z from noise prior
    Generate images G(z)
    Sample real images x
    Compute D loss: ℓ_D = Σ −log(1 − D(G(z))) − log(D(x))
    Take gradient descent step to update D
    # G update
    Sample noise vectors z from noise prior
    Generate images G(z)
    Compute G loss: ℓ_D = Σ −log(D(G(z)))
    Take gradient descent step to update G
end for
```

Revisiting Averages

Can use z to walk latent space

$G(\alpha z_1+(1-\alpha)z_2)$ for α in [0,1]





Remake of Slides by Isola, Freeman, Torralba, taken from Owens; diagram, duck + clownfish pictures from IFT.

[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

Conditioning on Things

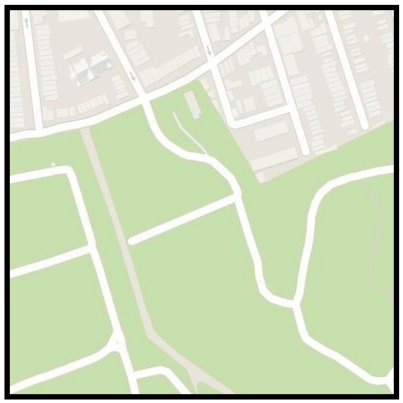
- Turning noise into pictures of things is all fun and good, but what if we want control over our synthetic images?

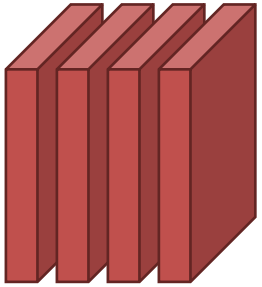
Conditional GANs (Pix2Pix)

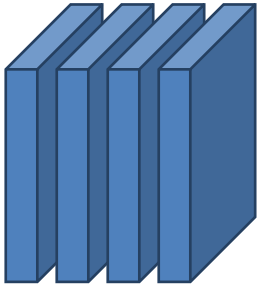
- Generator tries to make fake images – accepts **image** and makes an image
- Discriminator tries to identify fakes – outputs p(fake), potentially at each pixel

x G(·) G(x) D(·)

check each pixel if it's real or fake





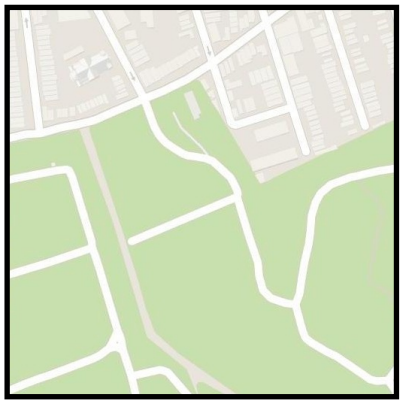


Real
v
Fake

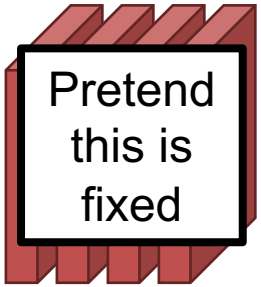
Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

Conditional GANs – Discriminator

x

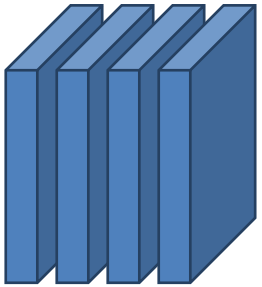


G(·)



G(x)

D(·)

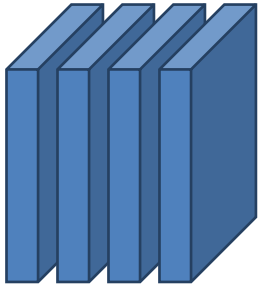


Want:
Fake

Want real outputs y to be high, fake output G(x) low

y (real image)

D(·)

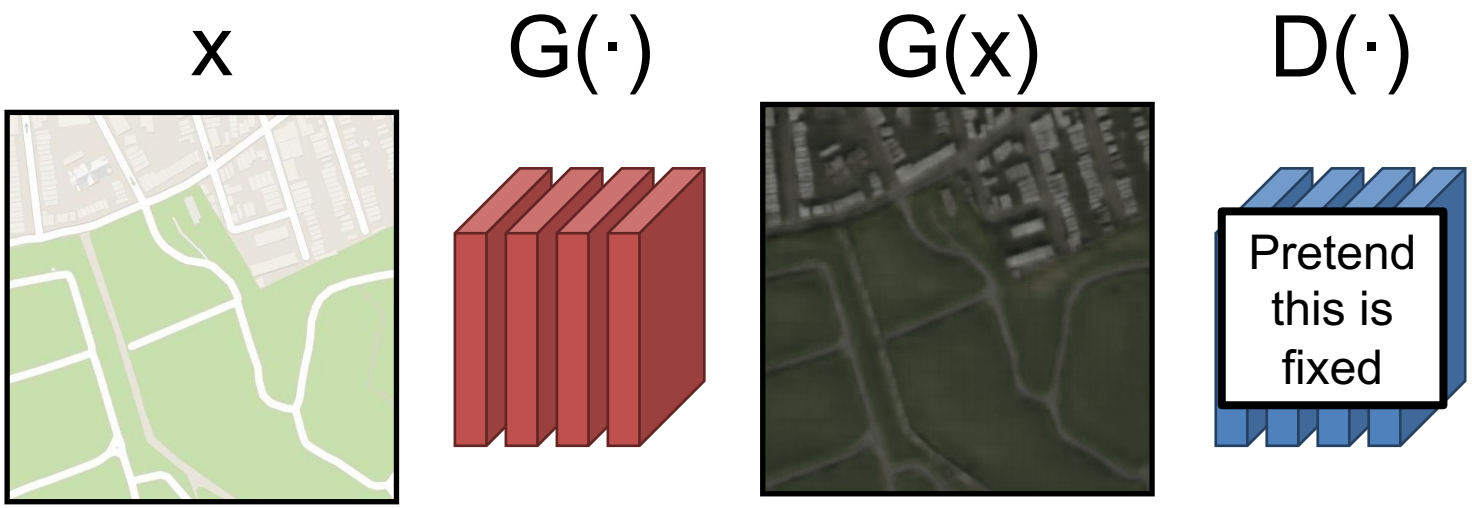


Want:
Real

$$\arg\max_{D} E_{z,x}[\log(1 - D(G(x))) + \log(D(y))]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

Conditional GANs – Generator



If you're the generator, want to make fakes that fool the discriminator into think they're real

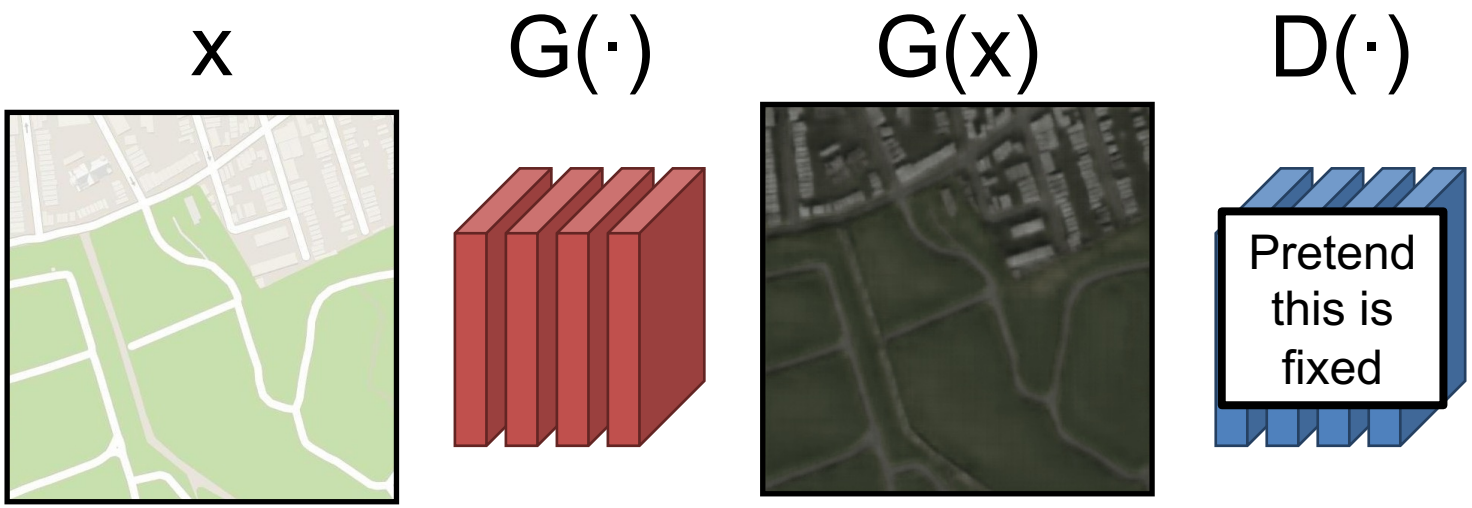
$$\arg\min_{G} E_{z,x}[\log(1 - D(G(x)))]$$

Same min/max game as before

$$\arg\min_{G}\max_{D} E_{z,x}[\log(1 - D(G(x))) + \log(D(y))]$$

Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

Conditional GANs – Generator



In practice, use alternating optimization using the improved G loss

$$\arg\min_{G} E_{z,x}[-\log\big(D(G(x))\big)]$$

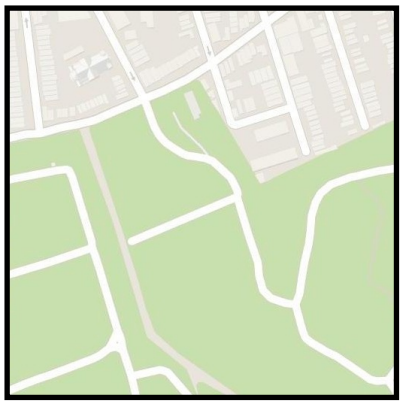
$$\arg\min_{D} E_{z,x}[-\log\Big(1-D\big(G(x)\big)\Big)-\log\big(D(y)\big)]$$

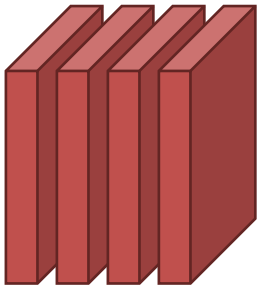
Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

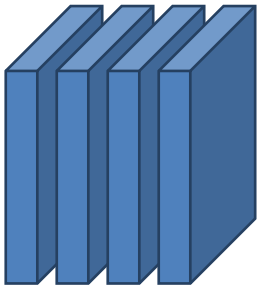
One Catch

- G can just output random good images.
- Solution – additional L1 loss

x G(·) G(x) D(·)







Real
v
Fake

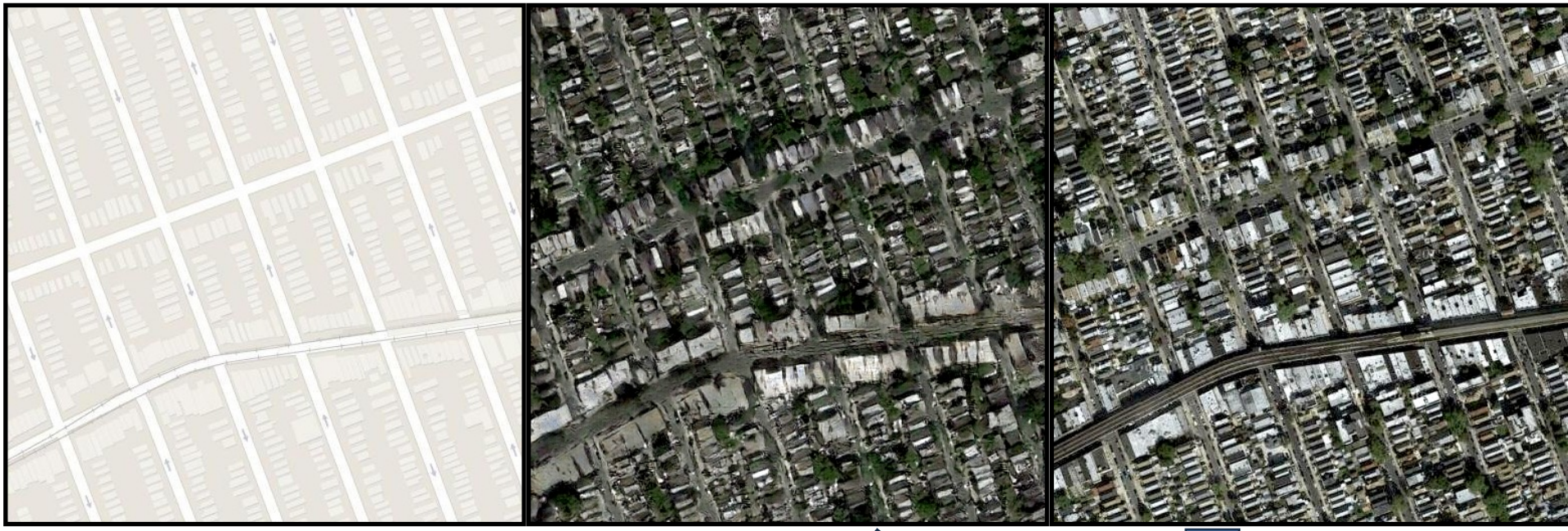
Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

GAN + L1 Loss

Input

Output

Groundtruth



Pixelwise L1 Loss

Remake of Slides by Isola, Freeman, Torralba, taken from Owens.

More Broadly

- Neural networks are lazy and will do precisely what you ask and no more
- You *have to* be careful what you ask them to do

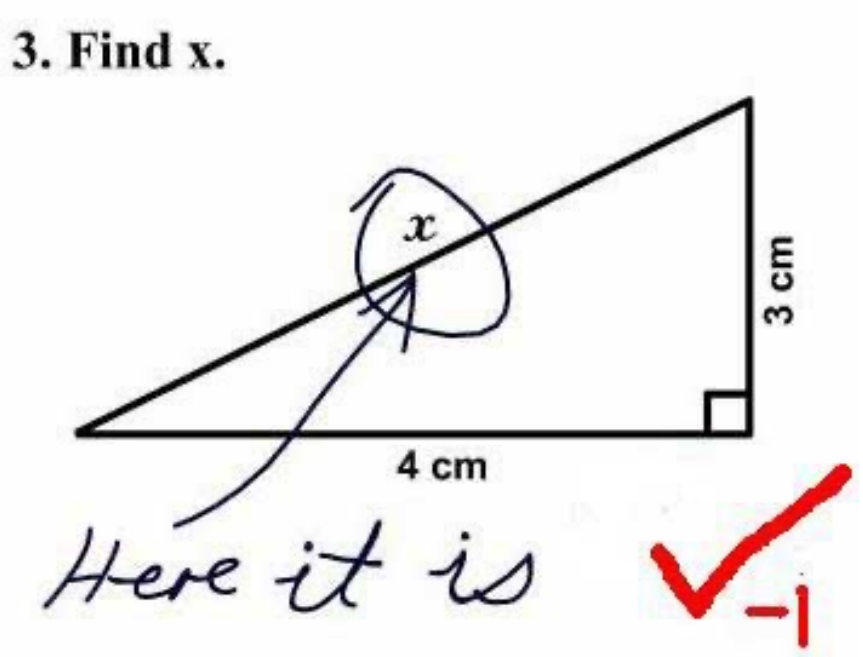


Photo credit: I found it at uregina.ca, but this is an old classic

Input

L1 loss

Why is it blurry junk? Hold that thought!

Slide by Isola, Freeman, Torralba, taken from Owens.

gan loss

Input

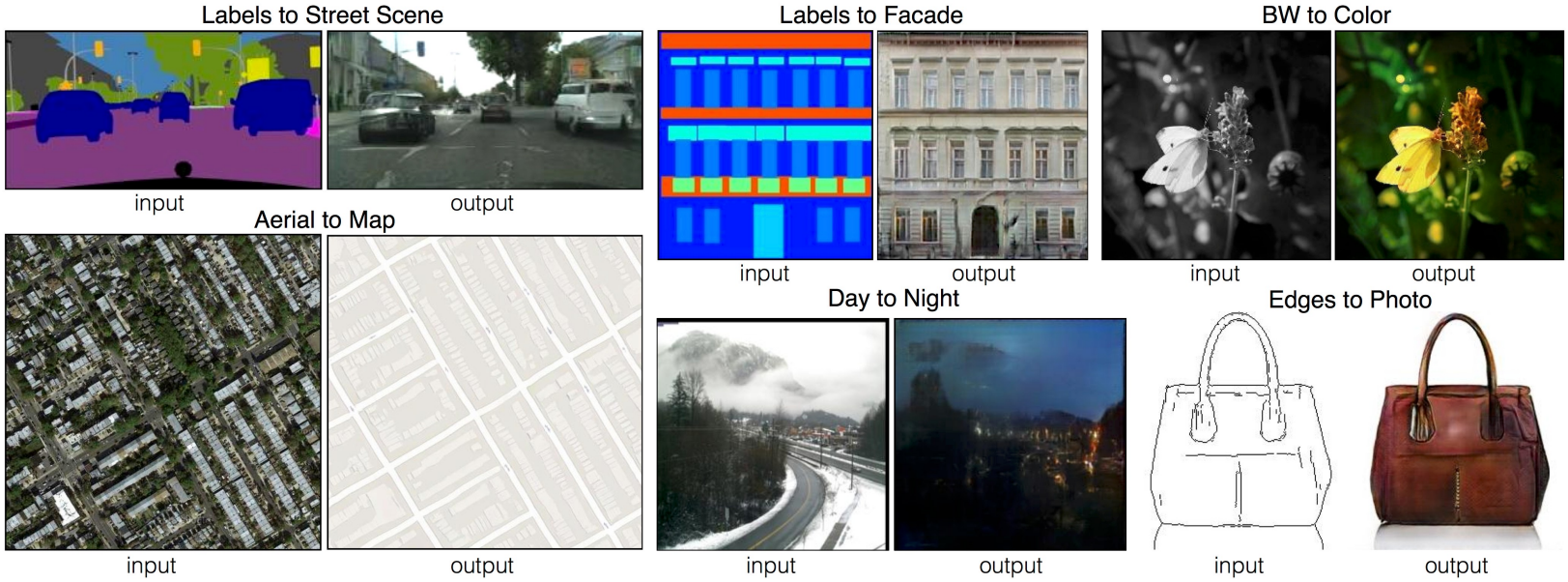
L1 loss + discriminator

Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

{isola,junyanz,tinghuiz,efros}@eecs.berkeley.edu



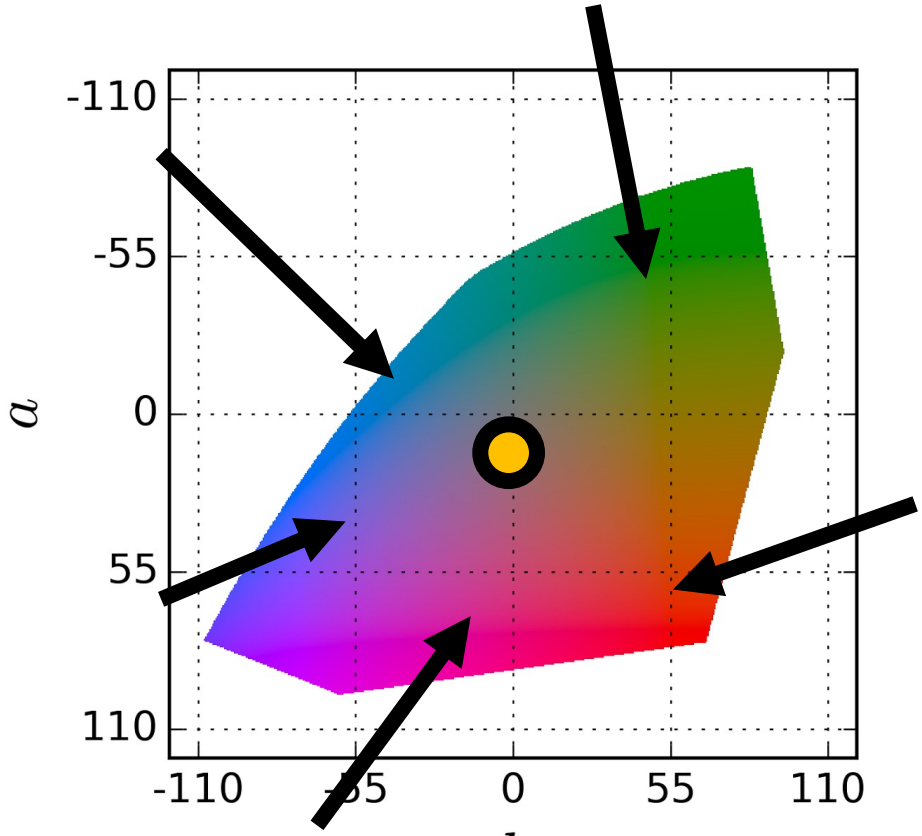
Let’s Talk About Blurry Pictures

What Color Is This Bird?

To make things more concrete: what color is this the pixel under this gold circle?

Example credit and diagrams: Richard Zhang

What Color Is This Bird?



L_2 : average

Many options. **What minimizes mean-squared error?**
What minimizes the L1 distance?

L_1 : median.

Example credit and diagrams: Richard Zhang

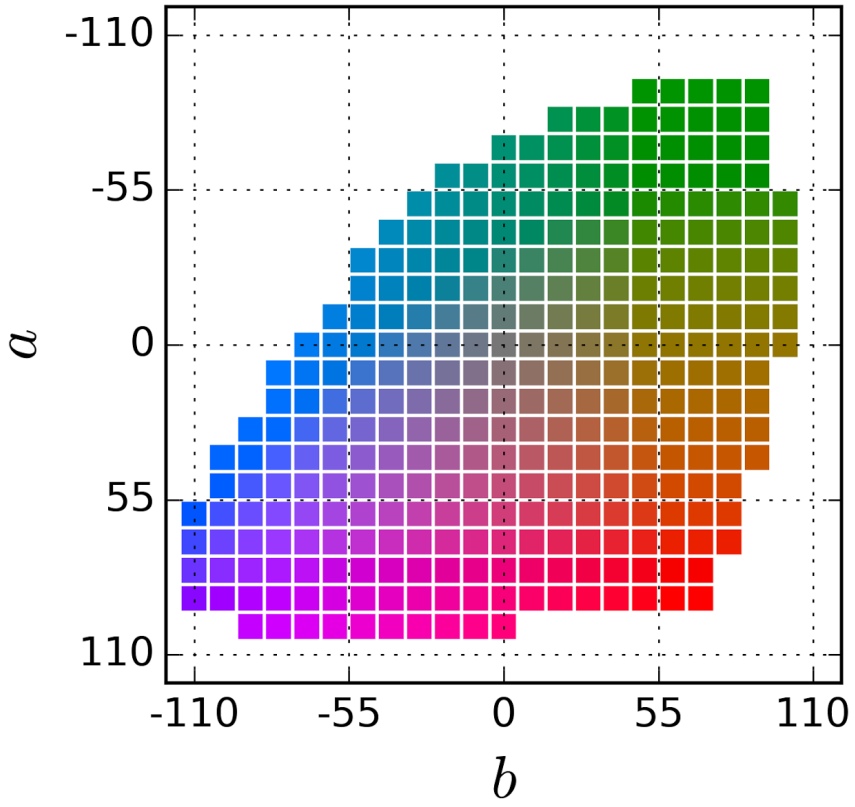
What Color Is This Bird?

Option 1: Discretize / quantize

Before learning: assign pixel nearest color index

After: convert index to value

Works because network can express its uncertainty.
Intuition: RGB prediction as a classification task

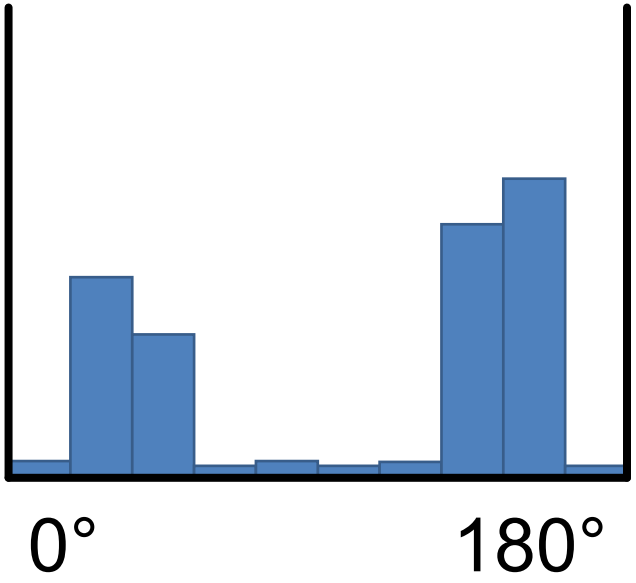


Example credit and diagrams: Richard Zhang

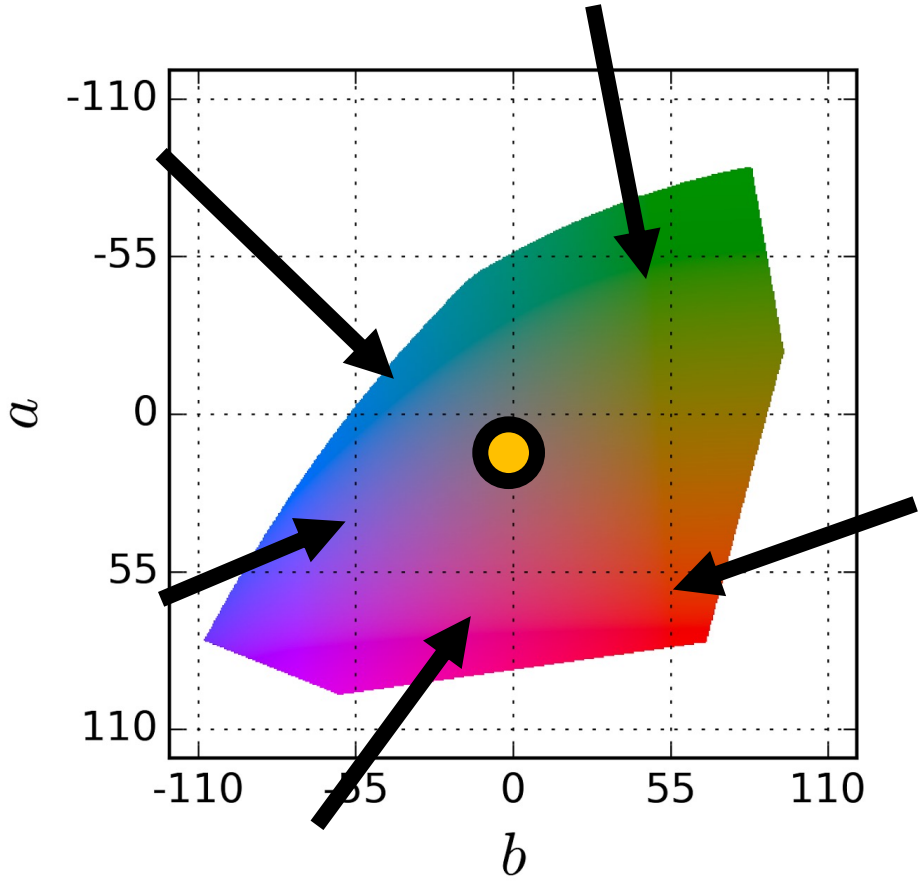
Discretized Values – Angles

Imagine predicting an angle from 0° to 180°. Having bins enables:

- Expressing bimodal distributions (e.g., either 30° or 150°)
- Getting a confidence from the prediction



Option 2 – GAN



GAN – the discriminator will prevent us from making birds grey or brown. **Why?**

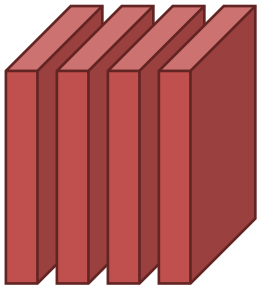
Example credit and diagrams: Richard Zhang

What to Take Away

- Be careful what you ask a deep net to solve.
- The objective you're asking it to solve bakes in assumptions
- Most solutions broken in one way or another
- Deep learning is not magic

Aside: Perceptual Losses

G(·)



G(x)

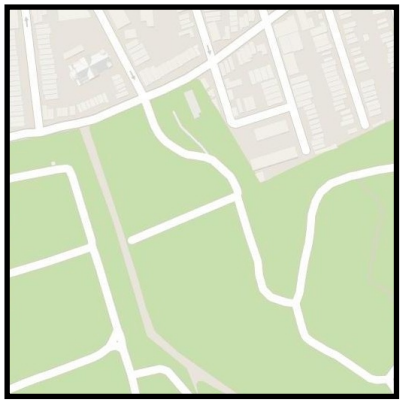
y

Conventionally, minimize distance in pixel space:

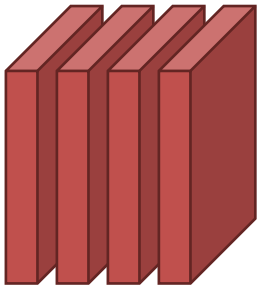
$$\|G(\boldsymbol{x}) - \boldsymbol{y}\|$$

See Johnson et al. Perceptual Losses for Real-Time Style Transfer and Super-Resolution, ECCV 2016.

Aside: Perceptual Losses

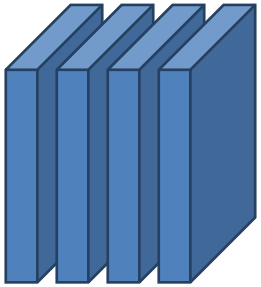


G(·)



G(x)

F(·)

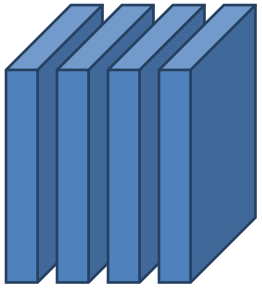


F(G(x))

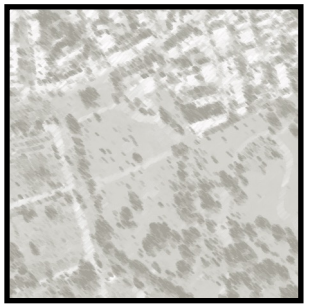
Instead measure distance after passing through pre-trained network

$$\|F(G(\boldsymbol{x})) - F(\boldsymbol{y})\|$$

y



F(y)

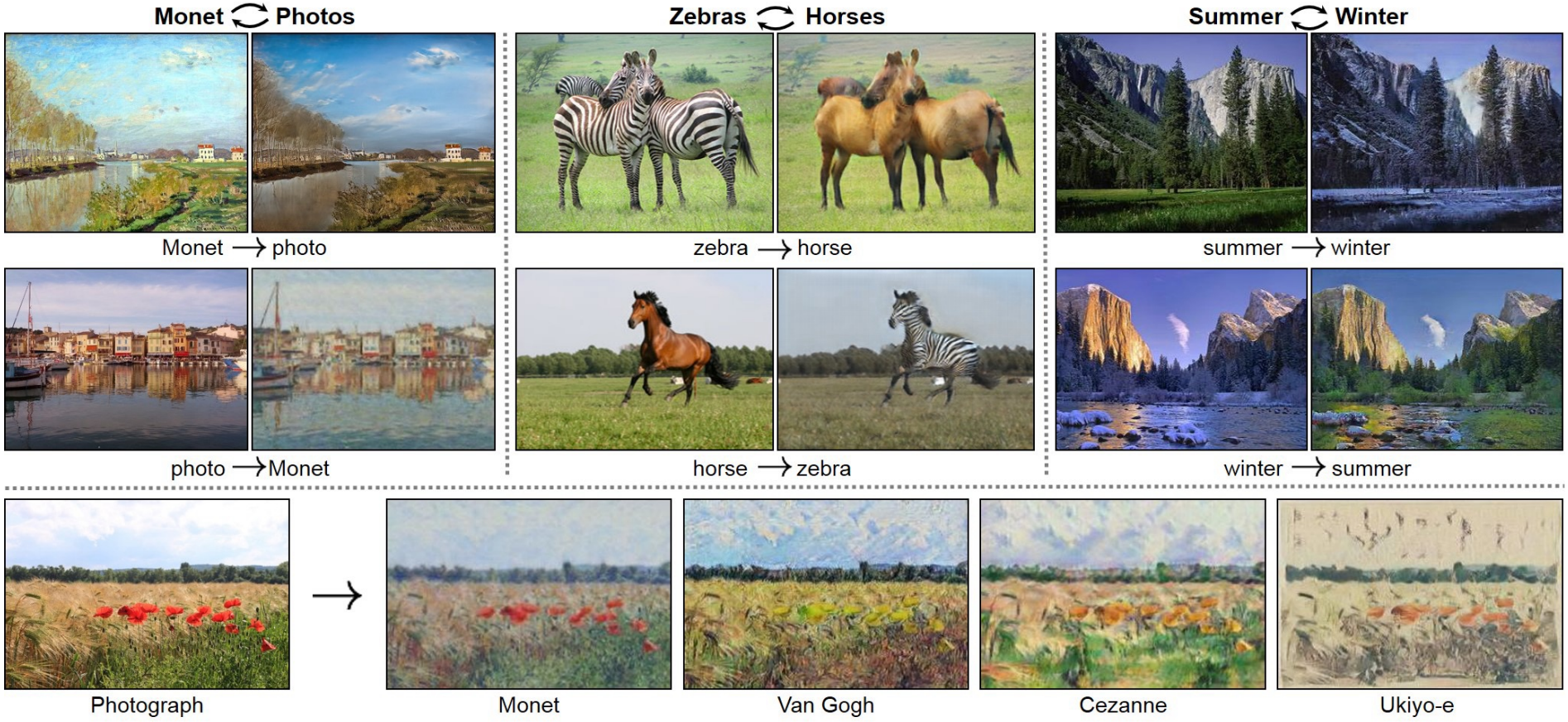


See Johnson et al. Perceptual Losses for Real-Time Style Transfer and Super-Resolution, ECCV 2016.

CycleGAN

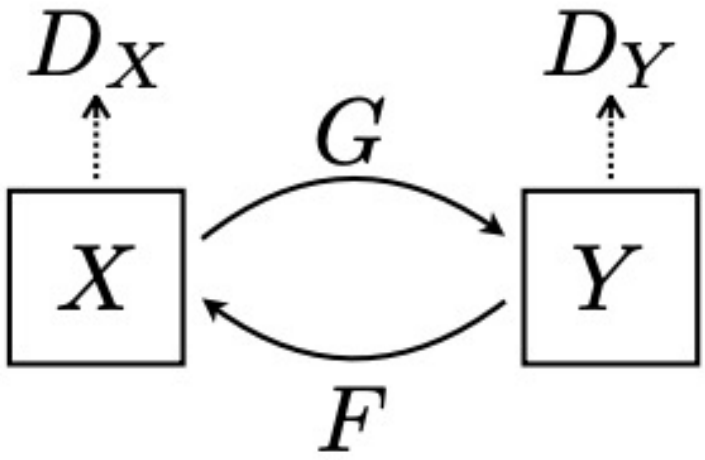
- Conditional GAN but don’t have paired data
- Only have two unpaired datasets

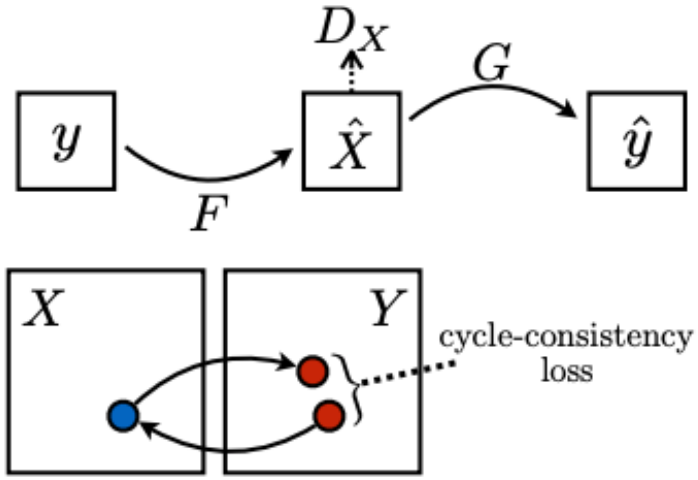
CycleGAN

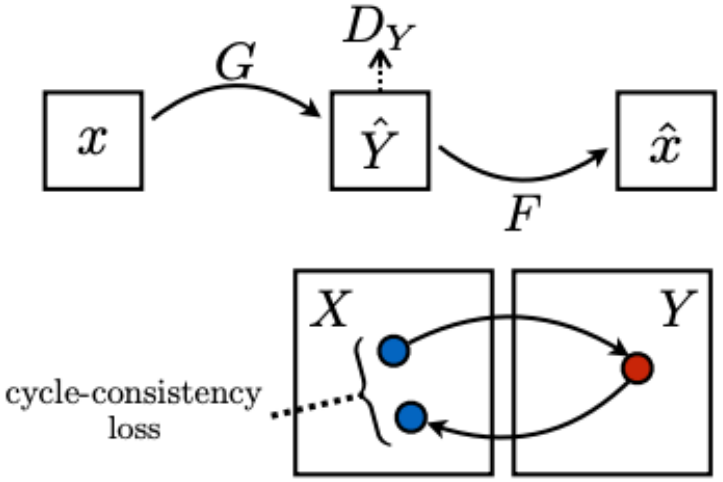


Training data: A set of images of style X + A set of images of style Y

Test: Given an image of style X, generate the same image in style Y


D_X
D_Y
G
X
Y
F


D_X
G
y
$\hat{X}$
$\hat{y}$
F
X
Y
cycle-consistency loss


D_Y
G
x
$\hat{Y}$
$\hat{x}$
F
X
Y
cycle-consistency loss

Unpaired Image Translation

Conditional generative model P(zebra images| horse images)

Zhu et al., 2017

Next Class

- Diffusion Models